Proceedings of the 9th Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE 2024)

Nobutaka Ono, Noboru Harada, Yohei Kawaguchi,
Woon-Seng Gan, Keisuke Imoto, Tatsuya Komatsu,
Qiuqiang Kong, and Irene Martin Morato (eds.)

October 23-25, 2024

DCASE2024 WORKSHOP

Citation:
Nobutaka Ono, Noboru Harada, Yohei Kawaguchi, Woon-Seng Gan, Keisuke Imoto, Tatsuya Komatsu, Qiuqiang Kong, Irene Martin Morato (eds.), Proceedings of the 9th Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE 2024), Oct. 2024.

DOI: https://doi.org/10.5281/zenodo.13871309

ISBN: 978-4-600-01518-3

Contents

Preface

This volume is a collection of the papers to be presented at the Detection and Classification of Acoustic Scenes and Events (DCASE) 2024 Workshop in Tokyo, Japan, on October 23-25, 2024. This ninth edition of the workshop is organized in conjunction with the DCASE Challenge, continuing the tradition from previous years. It will bring interested researcher from many universities and companies together and provide opportunities to exchange scientific ideas and opinions.

The DCASE 2024 Workshop is jointly organized by researchers at Tokyo Metropolitan University, NTT Corporation, Hitachi, Ltd., Nanyang Technological University, Doshisha University, LY Corporation, The Chinese University of Hong Kong, Tampere University, Apple Inc., Cochl, NEC Corporation, Mitsubishi Electric Research Laboratories, SONY, Ola Krutrim, CyberAgent, Inc., Google DeepMind, Toyohashi University of Technology, National Institute of Advanced Industrial Science and Technology (AIST), The University of Tokyo.

For this DCASE 2024 Workshop, 65 full papers were submitted. The submitted papers were assigned to four reviewers, receiving at least 3 reviews. Of these, 43 papers were accepted, including 15 oral and 28 poster presentations.

The organizing committee is also pleased to invite leading experts for keynote addresses:

- Nancy F. Chen
- Bourhan Yassin
- Jenelle Feather

The progress of the DCASE 2024 Workshop results from the hard work of many people whom we wish to extend a warm thanks to here, including the authors, the keynote speakers, and the reviewers, all without whom this DCASE 2024 Workshop would not exist. We also wish to thank the organizers and participants in the DCASE Challenge tasks.

Sponsorships support this edition of the workshop from Hitachi, Ltd., Samsung, LY Corpolation, NEC Corporation, Mitsubishi Electric Corporation, CyberAgent, Inc., Cochl, NTT Sonority, Inc., and grants from National Institute of Information and Communications Technology (NICT), Japan Science and Technology Agency (JST), SECOM Science and Technology Foundation, Kajima Foundation, and the Support Center for Advanced Telecommunications Technology Research (SCAT). We offer warm thanks for their valuable support of this workshop and the expanding topic area.

Nobutaka Ono
Noboru Harada
Yohei Kawaguchi
Woon-Seng Gan
Keisuke Imoto
Tatsuya Komatsu
Qiuqiang Kong
Irene Martin Morato

DCASE2024 Sponsors

DCASE2024 thanks the following sponsors for their valuable support.

Platinum Sponsor

HITACHI
Inspire the Next

Gold Sponsors

LY

\Orchestrating a brighter world
NEC

SAMSUNG

Silver Sponsors

CA
CyberAgent.

MITSUBISHI
ELECTRIC
Changes for the Better

MITSUBISHI ELECTRIC
RESEARCH LABORATORIES, INC

Bronze Sponsor

cochl.

Award Sponsor

NTT sonority

Grant Supporters

NICT National Institute of Information and Communications Technology

JST Japan Science and Technology Agency

公益財団法人
セコム科学技術振興財団
SECOM Science and Technology Foundation

公益財団法人
鹿島学術振興財団
THE KAJIMA FOUNDATION

SCAT

IMAD-DS: A DATASET FOR INDUSTRIAL MULTI-SENSOR ANOMALY DETECTION UNDER DOMAIN SHIFT CONDITIONS

Davide Albertini[1], *Filippo Augusti* [1], *Kudret Esmer*[2], *Alberto Bernardini*[2], *Roberto Sannino*[1]

[1] STMicroelectronics, Agrate Brianza (BG), Italy, {davide.albertini, filippo.augusti,roberto.sannino}@st.com
[2] Politecnico di Milano, DEIB, Milano (MI), Italy,
kudret.esmer@mail.polimi.it, alberto.bernardini@polimi.it

ABSTRACT

Industrial anomaly detection (AD) plays a critical role in maintaining the safety, efficiency and productivity of modern manufacturing and production processes. Despite the widespread adoption of IoT sensor boards in industry, there is still a lack of comprehensive multi-sensor and multi-rate datasets for AD that adequately account for domain shifts, i.e. variations in operational and environmental conditions that significantly affect AD performance. To address this gap, we present the Industrial Multi-sensor Anomaly Detection under Domain Shift Conditions (IMAD-DS) dataset. The IMAD-DS dataset comprises multi-sensor data from two scaled industrial machines: a robotic arm and a brushless motor, collected under different operating conditions to mimic real-world domain shifts, including speed and load changes. We also add different types of background noise to the audio data to simulate different environmental domain shifts. Benchmark testing with an autoencoder model show that AD performance decreases significantly with domain shifts, emphasizing the value of IMAD-DS for the development of robust multi-sensor AD systems.

Index Terms— Anomaly Detection, Sensor Fusion, Dataset, Domain Shift

1. INTRODUCTION

As modern industry grows in complexity and scale, the role of anomaly detection (AD) in machine monitoring and fault detection has increased significantly. This brings several benefits, such as increased safety, reduced impact on machine performance and higher productivity. Traditionally, industrial AD has relied on the experience of on-site technicians. While effective, this method is labor-intensive and often limited by the physical accessibility of some machine components. Therefore, the shift towards automated, data-driven methods such as machine learning and deep learning has gained momentum [1]. In this context, AD is framed as the task of automatically detecting abnormal conditions by learning only normal operating conditions.

A variety of physical variables such as vibration [2, 3, 4], temperature [5], pressure [6], and audio [7, 8, 9] can be used to detect anomalies in the industrial environment. However, with the widespread adoption of IoT boards it is now possible to simultaneously collect data from numerous sensors, providing a more comprehensive multi-modal description of machine operation. This data enables the development of more robust AD algorithms that take advantage of this richer description. Thus, the presence of multi-modal AD datasets becomes crucial for the development of the next generation of data-driven industrial AD systems.

Nevertheless, most existing industrial AD datasets primarily focus on single-sensor data, with only a few datasets covering multi-sensor scenarios. Notably, the Tennessee Eastman Process (TEP) models an industrial chemical process using a model-based simulator [10]. The HAI dataset captures data from a realistic industrial control system augmented with a hardware-in-the-loop simulator [11]. The CWRU Bearing dataset focuses on motor condition assessment [12]. Additionally, the Skoltech Anomaly Benchmark (SKAB) provides data from various machines captured using multiple sensors [13]. However, these datasets often overlook the inherent variability of real industrial environments that significantly affect the performance of AD systems [14, 15, 16, 17]. These deviations are often referred to as domain shifts and represent natural deviations in the distribution of normal data, which, however, make the automatic detection of anomalies more difficult.

The importance of accounting for domain shifts has recently been recognized in the field of audio-based anomaly detection, thanks in part to the contributions of the DCASE Task2 challenge and the availability of datasets that take this aspect into account, such as TOYADMOS2 [15], MIMII DUE [16] and MIMII-DG [17]. Introducing domain shifts into a dataset enables the development of more robust AD models and facilitates the development of domain adaptation and generalization techniques [17].

Inspired by the growing interest for AD in the presence of domain shifts, this paper introduces the Industrial Multi-sensor Anomaly Detection under Domain Shift Conditions (IMAD-DS) dataset. IMAD-DS comprises multi-sensor data from two scaled representations of industrial machines, namely a robotic arm and a brushless motor, collected under varying operational conditions to mimic real-world domain shifts, which include variations in operating speeds and loads. We also add different types of background noise to the audio signals to simulate different environmental domain shifts. Further, the dataset comprises sensors producing data with different sampling frequency, increasing the complexity with respect to single-rate multi-sensor datasets such as [10, 11, 13].

In addition to the dataset, we propose a deep learning model that enables multi-modal and multi-rate anomaly detection (AD) under domain shift conditions, serving as a benchmark to evaluate the dataset's usefulness. The model employs a fully connected autoencoder (AE) architecture that attempts to reconstruct multi-sensor data, yielding a reconstruction error which serves as an anomaly score metric for unsupervised AD. Results show that using multiple sensors is helpful for the task of AD, and also that performance decreases under domain shifts, underscoring the usefulness of the IMAD-DS dataset. The dataset is freely available for download at `https://zenodo.org/records/12636236`.

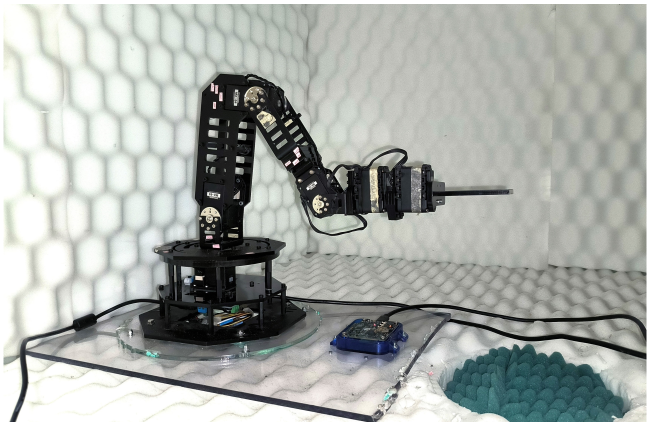


Figure 1: Robotic arm in the anechoic chamber, without weights. The IoT acquisition board is connected to the machine through the plexiglass base.

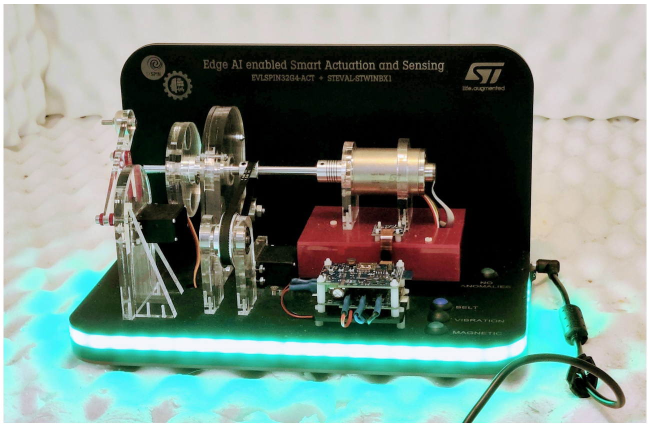

Edge AI enabled Smart Actuation and Sensing
EVLSPIN32G4-ACT + STEVAL-STWINBX1
life.augmented
NO ANOMALIES
BELT
VIBRATION
MAGNETIC

Figure 2: Brushless motor in the anechoic chamber. The IoT acquisition board is connected to the machine by screws that hold it on the plastic base.

Machine Name	**Domain Shift Parameter**	**Value for Source-Domain**	**Value for Target-Domain**
Robotic Arm	Attached loads of increasing weight	00, 10, 15, 20	25, 30, 35
	Factory background noise (SNR -4 dB)	A, C, D, F	B, E, G
Brushless Motor	Different rotation speeds [rpms]	1500, 1600, 1700, 1800, 1900, 2000, 2400, 2800, 3000	1000, 1100, 1200, 1300, 1400
	Factory background noise (SNR -4 dB)	A, C, D, F	B, E, G

Table 1: Domain shift configurations for the robotic arm and brushless motor in the IMAD-DS dataset. The table lists the different operational and environmental conditions used to create source and target domains. Numbers from 00 to 35 are indexes of increasing weights. Letters A to F index 7 background noise recordings from different real factories, all scaled to attain a SNR of -4 dB.

	Robotic Arm				**Brushless Motor**			
	Source Domain		**Target Domain**		**Source Domain**		**Target Domain**	
	Normal	**Anomaly**	**Normal**	**Anomaly**	**Normal**	**Anomaly**	**Normal**	**Anomaly**
Train	1812	0	27	0	1263	0	18	0
Test	116	116	116	116	78	78	78	78

Table 2: Number of samples for each class in source and target domains, further divided into normal and anomaly classes for the two machines.

2. DATASET OVERVIEW

The IMAD-DS dataset comprises multi-rate and multi-sensor data from two scaled representations of industrial machines, namely a robotic arm and a brushless motor. It contains both normal and abnormal multi-sensor data, which are also recorded under different operating conditions to account for domain shifts. The domain shifts considered in this dataset are divided into operational domain shifts, which are all the allowed machine working configurations, and environmental domain shifts, which are caused by changes in background noise. Anomalies are introduced by intentional disruptions to the normal behavior of the machine in question. IMAD-DS dataset consider the following machines.

Robotic Arm: The robotic arm is a scaled version of a robotic arm used to move silicon wafers in a factory, reproducing actual factory movements. The machine and its recording setup are shown in Fig. 1. Anomalies are created by loosening the screws at the arm's nodes, causing the typical spatial miscalibrations of such machines.

Brushless Motor: The brushless motor is a scaled representation of an industrial brushless motor, as shown in Fig. 2. Two anomalies are introduced: first, a magnet is moved closer to the motor load, causing oscillations by interacting with two symmetrical magnets on the load; second, a belt that rotates in unison with the motor shaft is tightened, creating mechanical stress.

To introduce domain shifts, various operating and environmental conditions are considered for each machine type. The robotic arm is recorded with seven different loads of increasing weight. In contrast, the brushless motor is recorded using 14 different operating voltages leading to various speeds. Both machines are also subjected to different background noises as environmental conditions. Combinations of these operating and environmental conditions divide each machine's dataset into two subsets, namely the source domain and the target domain. The source domain represents the original environment where a large number of training examples are available. In contrast, the target domain is characterized by a series of domain shifts where the availability of training data is severely limited and often restricted to few clips of target condition. The discrepancy between the source and target domains reflects a common problem in practice, where sufficient training data is often not available for the target domain. The domain shift configurations for both datasets are shown in Tab. 1.

As the dataset is tailored for unsupervised anomaly detection,

Sensor Type	Sample Rate	Part Number
Analog microphone	16 kHz	IMP23ABSU
3-axis Accelerometer	6.7 KHz	ISM330DHCX
3-axis Gyroscope	6.7 KHz	ISM330DHCX

Table 3: Sensors embedded on the STWIN.box IoT board and used for data acquisition.

this characteristic is also mirrored in the dataset's composition. Unsupervised AD systems exclusively use normal data for training, since acquiring a comprehensive set of real-world anomalies is challenging. Anomalous samples are included only in the test set to assess the system's capability to detect unknown anomalies. The composition of each machine dataset is detailed in Tab. 2.

3. RECORDING SETUP AND DATA PROCESSING

Multi-sensor data is collected using a STEVAL-STWINBX1 [18], an IoT Sensor Industrial Node from STMicroelectronics. In both machines, the sensor board and the machine lie on the same surface, allowing us to jointly characterize the machine's behavior in terms of audio, vibration, and rotations. The MEMS sensors used to capture these physical quantities are a microphone, an accelerometer, and a gyroscope, respectively. The actual sensors embedded on the sensor board and used to collect data are listed in Tab. 3 along with their respective sampling frequencies.

All recordings are conducted in a completely anechoic chamber, allowing precise control of the acoustic environment. This configuration not only enables detailed acoustic simulations, but also provides the flexibility to adjust the level of background noise to achieve the desired signal-to-noise ratio (SNR) and thus adjust the difficulty of the audio part of the AD task.

3.1. Processing of Audio Signals

The audio signals collected by the microphone are processed to simulate environmental domain shifts. For this purpose, machine noises are mixed with background noises recorded in real factories according to specific SNRs. In order to make the machine sounds and the background noise acoustically coherent, an acoustic simulation is performed to simulate a virtual acoustic environment in which sound sources, i.e. the background noises and the machine sounds, and a virtual microphone are present. Fig. 3 shows the configuration of the virtual acoustic environment in which the background noise sources are placed at the corners of a shoebox room with dimensions $10 \times 7.5 \times 4$ meters. The acoustic simulation is performed by employing the image source method (ISM) [19], which is used to calculate the room impulse response (RIR) of each virtual source and the virtual microphone, thus modeling the multi-path propagation of sound sources in the reverberant environment. In particular, the Pyroomacoustics library [20] is used to implement the ISM and to obtain the RIRs with a fixed reverberation time of $T60 = 0.5$ s.

Given the static nature of the acoustic environment under consideration, the RIRs are computed once for the entire dataset. The subsequent audio processing steps are as follows:

- Selection and Cropping: A background noise signal $\mathbf{n}$ is selected and cropped to match the length of the anechoic machine sound.
- Reverberation of Background Noise: The background noise signal $\mathbf{n}$ is convolved with the RIRs of the background noise emitters, producing the reverberated background noise signal at the virtual microphone $\mathbf{n}^{\text{rev}}$.
- Reverberation of Machine Sound: The machine sound $\mathbf{x}_{\text{mic}}^{\text{anech}}$ is convolved with its corresponding RIR, yielding $\mathbf{x}_{\text{mic}}^{\text{rev}}$.
- Scaling for SNR: The background noise $\mathbf{n}^{\text{rev}}$ is scaled to achieve the desired SNR using

$$\mathbf{n}^{\text{scaled}} = \mathbf{n}^{\text{rev}} \sqrt{\frac{P_{\mathbf{x}_{\text{mic}}^{\text{rev}}}}{10^{\text{SNR}/10} P_{\mathbf{n}^{\text{rev}}}}} \qquad (1)$$

where $P_{\mathbf{n}^{\text{rev}}}$ and $P_{\mathbf{x}_{\text{mic}}^{\text{rev}}}$ denote the power of the reverberated background noise and machine sound, respectively. SNRs are set according to Tab. 1.
- Final Cropping and Mixing: The signals $\mathbf{x}_{\text{mic}}^{\text{rev}}$ and $\mathbf{n}^{\text{scaled}}$ are cropped to the original machine sample length to remove the reverberation tail and are then mixed to produce the final audio sample used in the dataset.

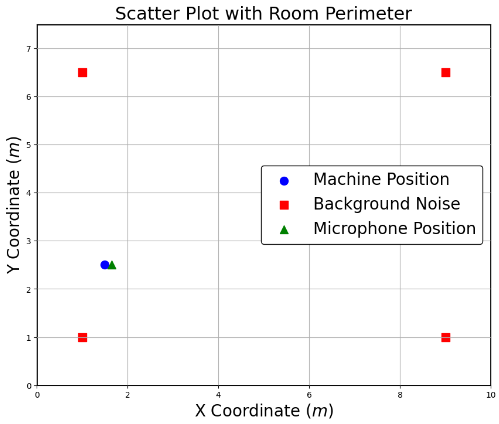


Figure 3: Acoustic environment simulated with the ISM. The red squares indicate the position of the background noise emitters, the blue circle the position of the machine sound emitter and the green triangle the position of the virtual microphone that senses the multi-path propagation of the sound sources.

Note that, in this work, we assume that the coupling of the machine with its surrounding environment is reflected only in the audio signals, as the acoustic coupling is more relevant than the others. The same setup used for the IMADS-DS dataset has also been used for generating audio files for the DCASE2024 task2 challenge *First-Shot Unsupervised Anomalous Sound Detection for Machine Condition Monitoring*.

4. EVALUATION AND BENCHMARK

To give an idea of the use and usefulness of the IMAD-DS dataset, we tested each machine sub-dataset on a simple baseline system. The Python codes for training, testing and creating the training and test data are available in the IMAD-DS dataset public repository.

4.1. Baseline

As a benchmark system, we use a fully-connected autoencoder (AE) that attempts to reconstruct an input vector consisting of all multi-rate, multi-sensor data related to the same temporal window. When an anomalous input is presented, a larger reconstruction error is expected, making the reconstruction error a valid anomaly score metric for unsupervised AD. The input of the baseline system consists of a column vector obtained by concatenating 100 ms windows of multi-sensor data. We denote each sensor data as $\mathbf{x}_s \in \mathbb{R}^{L_s C_s}$, where $s \in \mathcal{S} \triangleq \{\text{mic}, \text{acc}, \text{gyr}\}$ denotes a specific sensor, L_s is the number of samples in the 100 ms window given the sensor's sampling frequency, and C_s is the number of channels for that sensor (e.g., the accelerometer has x-, y- and z- axis components). Note that we stack all the sensor channels to form a single column vector of size $\sum_{s\in\mathcal{S}} L_s C_s$. Moreover, we apply a z-score normalization for each sensor channel, thereby obtaining the normalized sensor data $\tilde{\mathbf{x}}_s \in \mathbb{R}^{L_s C_s}$. Finally, the input of the AE is expressed as the concatenation of each normalized and stacked sensor data, i.e., $\mathbf{x} = [\tilde{\mathbf{x}}_{\text{mic}}^T, \tilde{\mathbf{x}}_{\text{acc}}^T, \tilde{\mathbf{x}}_{\text{gyr}}^T]^T$. The model encoder $E(\cdot|\theta_e)$, defined by trainable parameters θ_e, is composed of 3 fully connected (FC) layers with ReLU activation function [21], namely $FC(\sum_{s\in\mathcal{S}} L_s C_s, 2048, \text{ReLU})$, $FC(2048, 2048, \text{ReLU})$ and $FC(2048, 2048, \text{ReLU})$, and a bottleneck layer $FC(2048, 16, \cdot)$. The decoder $D(\cdot|\theta_d)$ mirrors the architecture of the encoder, with parameters θ_d. The model output is therefore the reconstructed input vector $\mathbf{x}' = D(E(\mathbf{x}|\theta_e)|\theta_d)$. The parameters of the encoder and decoder neural networks (i.e., $\theta = (\theta_e, \theta_d)$) are trained to minimize the loss function given as

$$\mathcal{L}(\theta_e, \theta_d) = \frac{1}{\sum_{s\in\mathcal{S}} L_s C_s} \|\mathbf{x} - D(E(\mathbf{x}|\theta_e)|\theta_d)\|_2^2 \qquad (2)$$

We trained the model with the Adam optimizer [22] with a learning rate of 10^{-4} and a batch size of 1024.

4.2. Results

To assess the AD performance of our benchmark model, we use the Area Under the Receiver Operating Characteristic Curve [23], i.e.,

$$\text{AUC} = \frac{1}{N_d^- N_d^+} \sum_{i=1}^{N_d^-} \sum_{j=1}^{N_d^+} \mathcal{H}\left(A_\theta(\mathcal{X}_j^+) - A_\theta(\mathcal{X}_i^-)\right), \qquad (3)$$

where $\mathcal{X}_i^-$ represents the ith normal data segment from the set of normal test data segments $\{\mathcal{X}_i^-\}_{i=1}^{N_d^-}$, and $\mathcal{X}_j^+$ is the jth anomalous data segment from the set of anomalous test data segments $\{\mathcal{X}_j^+\}_{j=1}^{N_d^+}$. Each data segment consists of several 100 ms windows. In this context, N_d^- and N_d^+ denote the total number of normal and anomalous test segments, respectively, with $d \in \{\text{Source}, \text{Target}, \text{Source} + \text{Target}\}$ specifying the domain under consideration. The anomaly score $A_\theta(\cdot)$ of each segment is the median reconstruction error of all inputs $\mathbf{x}$ within the segment. The function $\mathcal{H}(\cdot)$ outputs 1 if its input is positive, and 0 otherwise. Tab 4 summarizes the AD performance of the benchmark system. The columns labeled 'Source', 'Target', and 'S + T' present the AUC metrics for the source domain, the target domain, and the combined domain, respectively. The 'Overall' row displays the AUC calculated using the anomaly score from (2). To assess the benefits of using multi-sensor data over a single sensor setup, we also set all but one sensor data to zero. For instance, to evaluate the AD performance with only the microphone, we use as input to the AE the vector $\mathbf{x} = [\tilde{\mathbf{x}}_{\text{mic}}^T, \mathbf{0}^T, \mathbf{0}^T]^T$. For this configuration, (2) is evaluated on the subvectors corresponding to the microphone data only, i.e., $\mathbf{x}[: L_{\text{mic}} C_{\text{mic}}]$ for the input and $\mathbf{x}'[: L_{\text{mic}} C_{\text{mic}}]$ for the reconstructed output. The corresponding AUC metric is denoted in Tab. 4 as 'S-mic'. Moreover, we can also evaluate the single sensor AD performance when the others sensor data is present, i.e., when using as input to the AE $\mathbf{x} = [\tilde{\mathbf{x}}_{\text{mic}}^T, \tilde{\mathbf{x}}_{\text{acc}}^T, \tilde{\mathbf{x}}_{\text{gyr}}^T]^T$. For instance, to evaluate how AD performance of just the microphone is influenced by other sensors, we use again (2) on the microphone subvectors. The corresponding AUC is denoted in Tab. 4 as 'F-mic'. Results indicate that sensor-specific AUCs generally improve when incorporating data from other sensors, rather than relying solely on their own data. This suggests that multi-sensor data enhances performance even for single-sensor AD tasks. Furthermore, superior performance in the source domain over the target domain suggests domain shifts pose a challenge in the IMAD-DS dataset. In some instances, the 'S + T' AUC is lower than that of individual domains, as seen with 'F-acc' AUCs for the Robotic Arm dataset. This occurs when normal samples in the target domain have higher anomaly scores than anomalous samples in the source domain, leading the model to mistake domain changes for anomalies. For the Robotic Arm, the 'Overall' AUC exceeds individual sensors' AUCs in 'S + T' domain, which is not the case for the Brushless Motor dataset. This suggests that while sensor data fusion often aids AD, using (2) as an anomaly score does not guarantee that using all sensors always yield optimal performance. Therefore, exploring alternative multi-sensor methods is key to fully exploiting the potential of multi-sensor data.

Machine	Robotic Arm			Brushless Motor		
Domain	S + T	Source	Target	S + T	Source	Target
Overall	**91.62**	93.28	90.48	58.95	73.63	55.59
F-acc	90.49	**98.98**	**94.00**	**69.30**	**77.80**	**59.62**
S-acc	88.96	98.40	84.24	67.38	77.17	56.03
F-gyr	87.88	93.91	93.37	57.27	68.28	55.70
S-gyr	46.79	44.99	48.54	57.38	68.11	56.49
F-mic	66.31	73.27	63.18	54.19	58.83	49.27
S-mic	50.92	52.11	49.69	50.71	53.13	46.10

Table 4: Baseline AUC results, in percentage.

5. CONCLUSIONS

We presented IMAD-DS, a dataset developed to support the creation of domain adaptation and generalization strategies specifically tailored for multi-rate, multi-sensor AD systems in industrial settings. IMAD-DS includes both normal and abnormal operational data from two scaled versions of industrial machines, each collected under different operational scenarios to account for the variability in the domain. Our experiments with a fusing AE show improvements in AD when data from multiple sensors are included, compared to using data from a single sensor. Furthermore, we observe a decrease in AD efficacy due to domain shifts. This emphasizes the crucial role of IMAD-DS in the development of robust multi-rate multi-sensor systems for AD.

6. ACKNOWLEDGEMENTS

The authors whish to thank Alessandro Ilic Mezza for his support and suggestions throughout the development of the work.

7. REFERENCES

[1] L. Ruff, J. Kauffmann, R. Vandermeulen, G. Montavon, W. Samek, M. Kloft, T. Dietterich, and K. R. Müller, "A unifying review of deep and shallow anomaly detection," *Proceedings of the IEEE*, vol. PP, pp. 1–40, 02 2021.

[2] E. Carden, "Vibration based condition monitoring: A review," *Structural Health Monitoring*, vol. 3, pp. 355–377, 12 2004.

[3] M. Yu, D. Wang, and M. Luo, "Model-based prognosis for hybrid systems with mode-dependent degradation behaviors," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 1, pp. 546–554, 2014.

[4] G. S. Galloway, V. M. Catterson, T. Fay, A. Robb, and C. Love, "Diagnosis of tidal turbine vibration data through deep neural networks," vol. 3, no. 1, Jul. 2016.

[5] G. Lodewijks, W. Li, Y. Pang, and X. Jiang, "An application of the iot in belt conveyor systems," vol. 9864, 09 2016, pp. 340–351.

[6] R. F. Salikhov, Y. P. Makushev, G. N. Musagitova, L. U. Volkova, and R. S. Suleymanov, "Diagnosis of fuel equipment of diesel engines in oil-and-gas machinery and facilities," *AIP Conference Proceedings*, vol. 2141, no. 1, p. 050009, 08 2019. [Online]. Available: https://doi.org/10.1063/1.5122152

[7] H. Hojjati and N. Armanfard, "Self-supervised acoustic anomaly detection via contrastive learning," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 3253–3257.

[8] Y. Liu, J. Guan, Q. Zhu, and W. Wang, "Anomalous sound detection using spectral-temporal information fusion," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 816–820.

[9] K. Wilkinghoff, "Utilizing sub-cluster adacos for anomalous sound detection under domain shifted conditions," 2021.

[10] G. Manca, ""tennessee-eastman-process" alarm management dataset," 2020. [Online]. Available: https://dx.doi.org/10.21227/326k-qr90

[11] H.-K. Shin, W. Lee, J.-H. Yun, and H. Kim, *HAI 1.0: HIL-Based Augmented ICS Security Dataset*. USA: USENIX Association, 2020.

[12] W. A. Smith and R. B. Randall, "Rolling element bearing diagnostics using the case western reserve university data: A benchmark study," *Mechanical Systems and Signal Processing*, vol. 64-65, pp. 100–131, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0888327015002034

[13] I. D. Katser and V. O. Kozitsin, "Skoltech anomaly benchmark (skab)," https://www.kaggle.com/dsv/1693952, 2020.

[14] Y. Kawaguchi, K. Imoto, Y. Koizumi, N. Harada, D. Niizumi, K. Dohi, R. Tanabe, H. Purohit, and T. Endo, "Description and discussion on dcase 2021 challenge task 2: Unsupervised anomalous sound detection for machine condition monitoring under domain shifted conditions," 2021. [Online]. Available: https://arxiv.org/abs/2106.04492

[15] N. Harada, D. Niizumi, D. Takeuchi, Y. Ohishi, M. Yasuda, and S. Saito, "Toyadmos2: Another dataset of miniature-machine operating sounds for anomalous sound detection under domain shift conditions," 2021.

[16] R. Tanabe, H. Purohit, K. Dohi, T. Endo, Y. Nikaido, T. Nakamura, and Y. Kawaguchi, "Mimii due: Sound dataset for malfunctioning industrial machine investigation and inspection with domain shifts due to changes in operational and environmental conditions," 2021.

[17] K. Dohi, T. Nishida, H. Purohit, R. Tanabe, T. Endo, M. Yamamoto, Y. Nikaido, and Y. Kawaguchi, "Mimii dg: Sound dataset for malfunctioning industrial machine investigation and inspection for domain generalization task," 2022.

[18] "Stwin.box - sensortile wireless industrial node development kit." [Online]. Available: https://www.st.com/en/evaluation-tools/steval-stwinbx1.html

[19] J. Allen and D. Berkley, "Image method for efficiently simulating small-room acoustics," *The Journal of the Acoustical Society of America*, vol. 65, pp. 943–950, 04 1979.

[20] R. Scheibler, E. Bezzam, and I. Dokmanić, "Pyroomacoustics: A python package for audio room simulation and array processing algorithms," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 351–355.

[21] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?" in *2009 IEEE 12th International Conference on Computer Vision*, 2009, pp. 2146–2153.

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization." in *3rd International Conference for Learning Representations (ICLR), San Diego*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://dblp.uni-trier.de/db/conf/iclr/iclr2015.html#KingmaB14

[23] K. Dohi, T. Nishida, H. Purohit, R. Tanabe, T. Endo, M. Yamamoto, Y. Nikaido, and Y. Kawaguchi, "Mimii dg: Sound dataset for malfunctioning industrial machine investigation and inspection for domain generalization task," 05 2022.

ANGULAR DISTANCE DISTRIBUTION LOSS FOR AUDIO CLASSIFICATION

Antonio Almudévar[1]*, *Romain Serizel*[2], *Alfonso Ortega*[1]

[1] ViVoLab, Aragón Institute for Engineering Research (I3A), University of Zaragoza, Spain
[2] Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
almudevar@unizar.es

ABSTRACT

Classification is a pivotal task in deep learning not only because of its intrinsic importance, but also for providing embeddings with desirable properties in other tasks. To optimize these properties, a wide variety of loss functions have been proposed that attempt to minimize the intra-class distance and maximize the inter-class distance in the embeddings space. In this paper we argue that, in addition to these two, eliminating hierarchies within and among classes are two other desirable properties for classification embeddings. Furthermore, we propose the Angular Distance Distribution (ADD) Loss, which aims to enhance the four previous properties jointly. For this purpose, it imposes conditions on the first and second order statistical moments of the angular distance between embeddings. Finally, we perform experiments showing that our loss function improves all four properties and, consequently, performs better than other loss functions in audio classification tasks.

Index Terms— angular distance, audio classification, loss

1. INTRODUCTION

Classification is one of the main tasks to be solved with machine learning. In this task, there are typically high-dimensional elements and the goal is to decide to which class of a finite set each of these elements belongs. For this purpose, most of the solutions, particularly those based on deep learning, involve obtaining intermediate representations of reduced dimension of the elements to be classified. These representations are called embeddings and they can be considered as a summary of these elements containing the information that is relevant for classification. This problem is very popular not only because of its intrinsic importance, but also because it provides a simple way to obtain embeddings compared to other methods. Embeddings are useful for a multitude of tasks such as anomaly detection [1, 2], biometric recognition [3, 4], etc. The standard loss function to solve the classification task is the cross-entropy. As a secondary result of using this loss function, the embeddings of the different classes usually end up being somewhat separated. However, it is common to impose certain conditions directly on them due to two reasons: (i) this tends to improve the performance in the classification problem by guiding more the optimization [5, 6]; and (ii) it may be desirable for embeddings to have certain properties when used for a specific task other than classification [7, 8]. These conditions on embeddings are usually imposed through the loss function. Typically, a term is added to the cross-entropy or a modification is made to it.

In this paper we propose a loss function that is added to cross-entropy and we call it Angular Distance Distribution Loss because it imposes conditions on the first and second order statistical moments of the angular distances between embeddings in order to organize the embeddings in the space. Specifically, this organization consists of: (i) bringing embeddings of the same class closer, (ii) moving embeddings of different class away, (iii) minimizing the variation of the distances of the embeddings of the same class, and (iv) making the embeddings of a class equal in distance to the embeddings of any class. Traditionally, only the first two have been considered in the literature. However, in section 3 we formalize all four, arguing why they are all important. In addition, we reason how they relate to the statistical moments of the distances between embeddings. Furthermore, we propose an experimental framework with different Audio Classification datasets. In these experiments, on the one hand, we verify that our embeddings satisfy the properties described in the previous paragraph, so we verify that our loss function encourages the properties to be satisfied. On the other hand, we obtain a better accuracy than other loss functions that aim to establish conditions on the embeddings. Thus, we verify that the described properties translate into better classification performance. The details of these experiments are presented in the section 4 and can be replicated using the code in `https://github.com/antonioalmudevar/distance_distribution_loss`

2. RELATED WORK

Audio Classification consists of identifying to which class an audio belongs [9, 10]. In recent years it has received a lot of interest from the community [11, 12]. Solutions to this problem typically involve an embedding extractor followed by a small classifier net which are trained by minimizing cross-entropy. In many SOTA solutions the embedding extractor has a large number of parameters, so it is common to pre-train it with a large dataset and perform finetuning for the desired dataset. Although convolutional architectures has been widespread used [13–15], the most popular systems nowadays are transformer-based. These include Audio Spectrogram Transformer (AST) [16] and BEATs [17].

Loss Functions. It has been observed in different works that separating the embeddings of different classes often results in better performance in the classification task [5, 18–20]. Two loss functions that stand out are Focal Loss [7] and Orthogonal Projection Loss (OPL) [5], with which we compare our proposal.

*This work was supported by MCIN/AEI/10.13039/501100011033 under Grants PDC2021-120846-C41 & PID2021-126061OB-C44, and in part by the Government of Aragón (Grant Group T36 23R). This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 101007666.

3. PROPOSED METHOD

The problem we seek to solve in this paper is that of canonical classification, which has two characteristics: (i) all errors are considered equally critical; and (ii) all elements are considered equally similar to each other within a class. This means that intra-class and inter-class hierarchies are not desirable. In fact, the standard evaluation metric is accuracy, which considers all errors and correct predictions equally relevant. The presence of these hierarchies is desirable in some scenarios, but our goal is not to solve the latter.

3.1. Classification Solution Formulation

Let $\mathcal{D} = \{x^{(i)}, y^{(i)}\}_{i=1}^{N}$ be the dataset, where $x^{(i)}$ is each input and $y^{(i)}$ the label of $x^{(i)}$ and is a vector containing at each position j the probability that $x^{(i)}$ belongs to class j. The objective is to design a system that allows us to obtain a prediction of $y^{(i)}$ which we denote $\tilde{y}^{(i)}$. Typical deep learning classifier solutions consists of (i) an embeddings extractor f_θ, which provides the embedding as $z^{(i)} = f_\theta(x^{(i)}) \in \mathbb{R}^k$; and (ii) a classifier net g_ϕ, which gives the predictions as $\tilde{y}^{(i)} = g_\phi(z^{(i)}) \in \mathbb{R}^c$. Cross-entropy between y_i and $\tilde{y}_i$ is used as loss function, which we call $\mathcal{L}_{CE}$.

3.2. Desirable Properties of Embeddings

We explain below some desirable properties of embeddings for classification. In figs. 1 to 4 the dots correspond to low dimensional representations of the embeddings and different colors are used to indicate different classes.

- **Intra-class clustering**: The embeddings of the same class are close to each other in space. This has been shown to improve performance in classification and additional tasks.

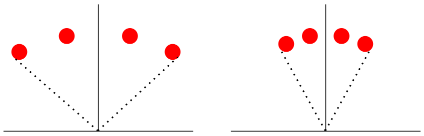


Figure 1: Low (left) and high (right) Intra-class clustering

- **Intra-class equidistance**: All the embeddings from the same class have approximately the same distance from each other. From a conceptual perspective, all the elements in a given class should be equally similar.

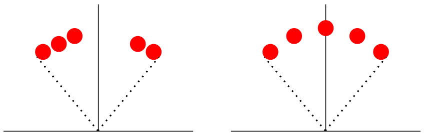


Figure 2: Low (left) and high (right) Intra-class equidistance

- **Inter-class separation**: Embeddings of different classes are far away from each other. This allows to take better advantage of all the space and, as a consequence, improves the performance in different tasks, especially when coupled with intra-class clustering.

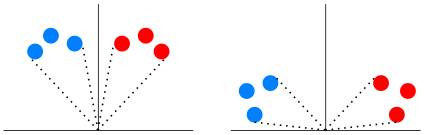


Figure 3: Low (left) and high (right) Inter-class separation

- **Inter-class equidistance**: All embeddings from different classes are approximately equally spaced from each other. This allows removing hierarchies between classes, which is conceptually desirable since all errors have the same penalty in the classical classification problem.

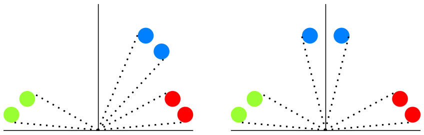


Figure 4: Low (left) and high (right) Inter-class equidistance

Traditionally, only intra-class clustering and inter-class separation have been considered as desirable properties. However, we also consider it convenient to have intra-class and inter-class equidistance, since these allow to avoid intra-class and inter-class hierarchies, respectively, which is desirable in the canonical classification problem, since all errors and correct predictions are equally critical. As a result, as we will see in section 4, maximizing these two improves the accuracy.

3.3. Angular Distance Distribution Loss

Having described the above properties and argued why they are desirable, we now present Angular Distance Distribution Loss, which encourages these properties. It imposes conditions on the first and second order statistical moments of the angular distances between embeddings. For now, we assume that the labels are hard, i.e. $y_k^{(i)} = 1$ for one k and 0 for the rest. With this idea, we can define the sets:

$$D_p = \left\{ d_c\left(z^{(i)}, z^{(j)}\right)^2 \,\middle|\, y^{(i)} = y^{(j)};\ i \neq j \right\} \quad (1)$$

$$D_n = \left\{ \left(1 - d_c\left(z^{(i)}, z^{(j)}\right)\right)^2 \,\middle|\, y^{(i)} \neq y^{(j)} \right\} \quad (2)$$

where $d_c(x, y) = 1 - x^T \cdot y$, which takes values in the interval $[0, 2]$, being 0 when the two vectors are proportional, 1 when they are orthogonal and 2 when they are opposites. Next, we define the following terms from the previous ones:

$$\mu_p = \frac{1}{|D_p|} \sum_{k \in D_p} k \quad (3)$$

$$\sigma_p = \sqrt{\frac{\sum_{k \in D_p} (k - \mu_p)^2}{|D_p| - 1}} \quad (4)$$

and μ_n and σ_n analogously for D_n. Each term can be related to one of the properties in 3.2 as follows:

- Minimizing μ_p implies boosting intra-class clustering, since it implies minimizing the average distance between embeddings of the same class.
- Minimizing σ_p implies promoting the intra-class equidistance, since we are reducing the variation of all the distances between embeddings of the same class.
- Minimizing μ_n implies boosting the inter-class separation, since we are promoting the embeddings of different classes to be orthogonal
- Minimizing σ_n implies favoring the inter-class equidistance, since we are reducing the variation between embedding distances of different classes.

With all this, we define our loss function to minimize ADD as:

$$L_{ADD} = \lambda_\mu^p \mu_p + \lambda_\sigma^p \sigma_p + \lambda_\mu^n \mu_n + \lambda_\sigma^n \sigma_n \tag{5}$$

where $\boldsymbol{\lambda} = \{\lambda_\mu^p, \lambda_\sigma^p, \lambda_\mu^n, \lambda_\mu^n\}$ are hyperparameters. In section 4 we explore how each of these terms separately affects the accuracy and distribution of embeddings in space.

Finally, the loss function of our system is as follows:

$$\mathcal{L} = \mathcal{L}_{CE} + \mathcal{L}_{ADD} \tag{6}$$

3.4. Soft Labels Adaptation

In some scenarios, the labels used to optimize our model are soft, i.e., they represent the probability that an element belongs to each class instead of considering that an element belongs to a single class [21]. One of the main causes of having soft labels is the use of data augmentation techniques such as mixup [22]. As mixup is widely used in audio classification [23], we propose a modification of our loss function to deal with soft labels.

When we have soft labels, it is still important to maximize intra-class clustering and intra-class equidistance, since we are interested that elements belonging to the same class should be close and at a similar distance from each other. However, inter-class separation must be reinterpreted, so that it would be desirable that if $y^{(i)}$ is more similar to $y^{(j)}$ than to $y^{(k)}$, then $z^{(i)}$ should be closer to $z^{(j)}$ than to $z^{(k)}$, and vice versa. For this, we must strive that $d_c\left(z^{(i)}, z^{(j)}\right) = d_c\left(y^{(i)}, y^{(j)}\right)$ for each pair i, j. To modify the loss function, we first define:

$$\mathcal{L}_\mu = \frac{1}{N_B} \sum_{i \in B} \sum_{j \neq i} \left(d_c\left(y^{(i)}, y^{(j)}\right) - d_c\left(z^{(i)}, z^{(j)}\right) \right)^2 \tag{7}$$

where $N_B = |B|(|B| - 1)$ is the number of pairs in a batch. Optimizing $\mathcal{L}_\mu$ we manage to jointly maximize intra-class clustering and inter-class separation. In fact, we note that we do not lose generality with respect to the hard scenario, since $d_c\left(y^{(i)}, y^{(j)}\right)$ holds 0 if $y^{(i)} = y^{(j)}$ and 1 otherwise and, therefore, optimizing $\mathcal{L}\mu$ is equivalent to optimizing $\lambda_\mu^p \mu_p + \lambda_\mu^n \mu_n$ with $\lambda_\mu^n = \frac{|D_n|}{|D_p|} \lambda_\mu^p$. In addition, since elements do not belong to a single class, it does not make sense to maximize the inter-class equidistance. Thus, when we have soft labels, we define the ADD as:

$$L_{ADD}^{soft} = \lambda_\mu \mathcal{L}_\mu + \lambda_\sigma^p \sigma_p \tag{8}$$

4. EXPERIMENTS

4.1. Datasets

Environmental Sound Classification (ESC-50) [24] contains 2,000 5-second ambient sound recordings annotated with 5 classes, so that each audio belongs to a single class. In our experiments we follow the standard 5-fold cross-validation to evaluate our systems.
Speech Commands V2 (KS2) [25] is composed of 105,829 clips of 1-second spoken keywords annotated with 35 word classes. It is officially divided into 84,843, 9,981 and 11,005 clips for training, test and validation, respectively.
IEMOCAP (ER) [26] contains about 12 hours of speech with four different emotions. We use the standard 5-fold cross-validation proposed in [27] for evaluation.

Table 1: Hyperparam. per embeddings extractor and dataset

	AST			BEATs		
	ESC	KS2	ER	ESC	KS2	ER
Window type	Hanning			Povey		
Freq. Mask	24	48	24	0		
Time Mask	96	48	96	0		
Mixup λ	0	0.5	0	0	0.5	0
Epochs	25			30		
Batch Size	32			16		
Optimizer	AdamW			Adam		
Learning rate	7e-4	6e-5	7e-4	8e-6	1e-4	8e-6
Momentum	$\boldsymbol{\beta} = \{0.9, 0.98\}$			$\boldsymbol{\beta} = \{0.95, 0.999\}$		
Weight Decay	1e-2			5e-6		

4.2. Embeddings Extractors Architectures

Audio Spectrogram Transformer (AST) [16] is the first to use Transformer type architectures for audio. The original AST model is pre-trained on Imagenet [28] and Audioset [9]. We fine-tune it for each scenario.
Bidirectional Encoder representation from Audio Transformers (BEATs) [17] is a pre-training framework for learning representations from Audio Transformers, in which an acoustic tokenizer and a self-supervised audio model are optimized. We use the original pre-trained model with Audioset and finetune for each scenario.

4.3. Hyperparameters

For all our systems we use the audio signals at 16kHz. The input to the systems are 128 mel-spectrograms coefficients computed on 25 ms windows every 10 ms. We normalize the mean and standard deviation to 0 and 0.5, respectively. Some hyperparameters vary between scenarios. These details can be found in table 1 and are inspired by the experiments in the original papers, with slight modifications due to computational limitations.

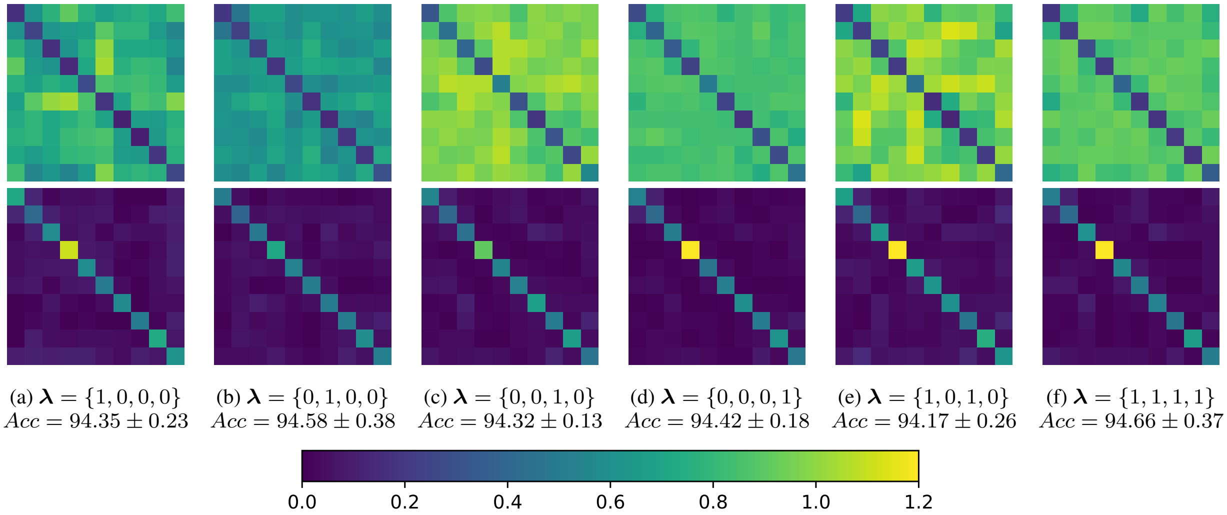
4.4. Ablation Study on each term of the ADD

We are going to analyze the influence of each of the terms of $\mathcal{L}_{ADD}$. First, we want to see if the hypotheses outlined in section 3.3 about the relationship between each ADD term and the properties of 3.2 hold. Second, we want to analyse the impact of each particular term in the ADD and their combination on the accuracy. For this, we train four classifiers, each with one of the elements of $\boldsymbol{\lambda}$ equal to 1 and the rest equal to zero. Third, we want to test whether optimizing intra-class and inter-class equidistance provides an advantage despite already optimizing intra-class clustering and inter-class separation. To do so, we train two classifiers: one with $\boldsymbol{\lambda} = \{1, 0, 1, 0\}$ and another with $\boldsymbol{\lambda} = \{1, 1, 1, 1\}$ and compare them. The dataset to be classified is ESC-50 and the embedding extractor used an AST in all cases. In figure 5 we present the mean and coefficient of variation of the d_c between the embeddings of 10 pairs of classes and the accuracy calculated for all the classes.

- In figure 5a we see that the distance between embeddings of the same class is in general the minimum in mean, i.e. the intra-class clustering is the maximum.
- In figure 5b we observe that the distances between the embeddings of the same class is the least spread, which means that the intra-class equidistance is the highest.

Table 2: Accuracy for the different Datasets, Embeddings Extractors and Loss Functions

	ESC-50		KS2		ER	
	AST	BEATs	AST	BEATs	AST	BEATs
Cross-entropy	93.97 ± 0.21	91.05 ± 0.41	92.05 ± 0.04	88.94 ± 0.13	59.91 ± 0.60	61.66 ± 0.31
Focal Loss [7]	94.40 ± 0.36	91.10 ± 0.49	-	-	60.79 ± 0.16	62.17 ± 0.05
OPL [5]	94.11 ± 0.37	91.50 ± 0.20	-	-	60.53 ± 0.42	$\mathbf{63.06 \pm 0.32}$
ADD ($\boldsymbol{\lambda} = \{1, 1, 1, 1\}$)	$\mathbf{94.68 \pm 0.09}$	$\mathbf{92.22 \pm 0.06}$	$\mathbf{97.54 \pm 0.06}$	$\mathbf{90.49 \pm 0.16}$	$\mathbf{61.30 \pm 0.38}$	62.73 ± 0.17



(a) $\boldsymbol{\lambda} = \{1, 0, 0, 0\}$ $Acc = 94.35 \pm 0.23$

(b) $\boldsymbol{\lambda} = \{0, 1, 0, 0\}$ $Acc = 94.58 \pm 0.38$

(c) $\boldsymbol{\lambda} = \{0, 0, 1, 0\}$ $Acc = 94.32 \pm 0.13$

(d) $\boldsymbol{\lambda} = \{0, 0, 0, 1\}$ $Acc = 94.42 \pm 0.18$

(e) $\boldsymbol{\lambda} = \{1, 0, 1, 0\}$ $Acc = 94.17 \pm 0.26$

(f) $\boldsymbol{\lambda} = \{1, 1, 1, 1\}$ $Acc = 94.66 \pm 0.37$

Figure 5: Mean (top row) and coefficient of variation (bottom row) of the d_c values between embeddings of 10 classes of ESC-50. The accuracy given is calculated for the 50 classes. Coefficient of variation is defined as $\frac{\sigma}{\mu}$ and is used here instead of σ because it normalizes the variation by normalizing by the mean, which changes depending on $\boldsymbol{\lambda}$, so it represents better intra-class equidistance.

- In figure 5c we contemplate that the inter-class distance or separation is the maximum in mean.
- In figure 5d we find that the distances between embeddings of different classes are similar regardless of the class pairs, thus achieving a higher inter-class equidistance.
- In figure 5e we see that if we only optimize intra-class clustering and inter-class separation, the distances between different pairs of embeddings of different classes are very far from each other. In addition, there are classes for which embeddings are closer to each other than for other classes.
- In figure 5f we obtain embeddings that do not satisfy each property as well as when we try to optimize them separately, but with a good balance between all of them.

Finally, the best accuracy obtained is for $\boldsymbol{\lambda} = \{1, 1, 1, 1\}$, that is, when we optimize all four properties together. This leads us to believe that all these properties have an influence in achieving a higher accuracy. In addition, we see that the properties that separately have most positively influence accuracy are inter-class and intra-class equidistance.

4.5. Quantitative Results

We have found in the previous section that our loss function allows us to meet the properties that we consider desirable. Moreover, we have verified that for the analyzed scenario, the accuracy when the four properties are optimized jointly is better than when they are optimized separately. Here we perform a more extensive study in which we compare in terms of accuracy the ADD with other loss functions with good performance. We do not use Focal Loss and OPL for KS2, as these do not support soft labels and we use mixup for this dataset. We have performed all the experiments three times and we provide the mean and standard deviation of the accuracy. In table 2 we can see that the results of our loss function is superior to the rest except in one case. This suggests that, in general, the described properties are desirable to improve accuracy and that the ADD function performs superiorly in different scenarios.

5. CONCLUSIONS AND FUTURE WORK

In this paper we have presented four properties for embeddings of a classifier arguing why we consider these properties to be desirable. In addition, we have designed Angular Distance Distribution Loss, a loss function that is intended to allow us to obtain each of these properties. First, we have verified that, indeed, our loss function allows us to obtain emebddings that satisfy these properties separately and jointly. Subsequently, we have observed, for a given scenario, that the performance in terms of accuracy is better when all four properties are satisfied jointly than separately. Finally, we have found for different datasets and architectures that the fact that our embeddings satisfy these properties translates into better accuracy than other relevant loss functions in the literature. This validates the hypothesis about the importance of these properties to improve accuracy. We believe that the properties described in this work may be desirable also for other applications, such as anomaly detection or biometric recognition. Thus, experiments testing the ADD in these fields can be developed in the future

6. REFERENCES

[1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[2] Y. Kawaguchi, K. Imoto, Y. Koizumi, N. Harada, D. Niizumi, K. Dohi, R. Tanabe, H. Purohit, and T. Endo, "Description and discussion on dcase 2021 challenge task 2: Unsupervised anomalous sound detection for machine condition monitoring under domain shifted conditions," *arXiv preprint arXiv:2106.04492*, 2021.

[3] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4690–4699.

[4] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "Cosface: Large margin cosine loss for deep face recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5265–5274.

[5] K. Ranasinghe, M. Naseer, M. Hayat, S. Khan, and F. S. Khan, "Orthogonal projection loss," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 12 333–12 343.

[6] A. Almudévar, A. Ortega, L. Vicente, A. Miguel, and E. Lleida, "Variational Classifier for Unsupervised Anomalous Sound Detection under Domain Generalization," in *Proc. INTERSPEECH 2023*, 2023, pp. 2823–2827.

[7] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[8] Y. Zeng, H. Liu, L. Xu, Y. Zhou, and L. Gan, "Robust anomaly sound detection framework for machine condition monitoring," DCASE2022 Challenge, Tech. Rep., July 2022.

[9] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 776–780.

[10] A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen, "Dcase 2017 challenge setup: Tasks, datasets and baseline system," in *DCASE 2017-Workshop on Detection and Classification of Acoustic Scenes and Events*, 2017.

[11] N. Turpault, R. Serizel, A. P. Shah, and J. Salamon, "Sound event detection in domestic environments with weakly labeled data and soundscape synthesis," in *Workshop on Detection and Classification of Acoustic Scenes and Events*, 2019.

[12] F. Ronchini, R. Serizel, N. Turpault, and S. Cornell, "The impact of non-target events in synthetic soundscapes for sound event detection," *arXiv preprint arXiv:2109.14061*, 2021.

[13] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, *et al.*, "Cnn architectures for large-scale audio classification," in *2017 ieee international conference on acoustics, speech and signal processing (icassp)*. IEEE, 2017, pp. 131–135.

[14] E. Cakır, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1291–1303, 2017.

[15] R. Serizel, N. Turpault, H. Eghbal-Zadeh, and A. P. Shah, "Large-scale weakly labeled semi-supervised sound event detection in domestic environments," *arXiv preprint arXiv:1807.10501*, 2018.

[16] Y. Gong, Y.-A. Chung, and J. Glass, "Ast: Audio spectrogram transformer," *arXiv preprint arXiv:2104.01778*, 2021.

[17] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, Z. Chen, and F. Wei, "Beats: Audio pre-training with acoustic tokenizers," *arXiv preprint arXiv:2212.09058*, 2022.

[18] G. Sun, S. Khan, W. Li, H. Cholakkal, F. S. Khan, and L. Van Gool, "Fixing localization errors to improve image classification," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16*. Springer, 2020, pp. 271–287.

[19] X. Zhang, R. Zhao, Y. Qiao, and H. Li, "Rbf-softmax: Learning deep representative prototypes with radial basis function softmax," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI 16*. Springer, 2020, pp. 296–311.

[20] I. Sheth and S. Ebrahimi Kahou, "Auxiliary losses for learning generalizable concept-based models," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[21] I. Martín-Morató, M. Harju, P. Ahokas, and A. Mesaros, "Strong labeling of sound events using crowdsourced weak labels and annotator competence estimation," in *Proc. IEEE Int. Conf. Acoustic., Speech and Signal Process. (ICASSP)*, 2023.

[22] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[23] Z. Zhang, S. Xu, S. Cao, and S. Zhang, "Deep convolutional neural network with mixup for environmental sound classification," in *Chinese conference on pattern recognition and computer vision (prcv)*. Springer, 2018, pp. 356–367.

[24] K. J. Piczak, "Esc: Dataset for environmental sound classification," in *Proceedings of the 23rd ACM international conference on Multimedia*, 2015, pp. 1015–1018.

[25] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *arXiv preprint arXiv:1804.03209*, 2018.

[26] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan, "Iemocap: Interactive emotional dyadic motion capture database," *Language resources and evaluation*, vol. 42, pp. 335–359, 2008.

[27] S.-w. Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin, *et al.*, "Superb: Speech processing universal performance benchmark," *arXiv preprint arXiv:2105.01051*, 2021.

[28] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

HETEROGENEOUS SOUND CLASSIFICATION WITH THE *BROAD SOUND* TAXONOMY AND DATASET

Panagiota Anastasopoulou, Jessica Torrey, Xavier Serra, Frederic Font

Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain
panagiota.anastasopoulou@upf.edu, jessica@jessicatorrey.com, xavier.serra@upf.edu, frederic.font@upf.edu

ABSTRACT

Automatic sound classification has a wide range of applications in machine listening, enabling context-aware sound processing and understanding. This paper explores methodologies for automatically classifying heterogeneous sounds characterized by high intra-class variability. Our study evaluates the classification task using the Broad Sound Taxonomy, a two-level taxonomy comprising 28 classes designed to cover a heterogeneous range of sounds with semantic distinctions tailored for practical user applications. We construct a dataset through manual annotation to ensure accuracy, diverse representation within each class and relevance in real-world scenarios. We compare a variety of both traditional and modern machine learning approaches to establish a baseline for the task of heterogeneous sound classification. We investigate the role of input features, specifically examining how acoustically derived sound representations compare to embeddings extracted with pre-trained deep neural networks that capture both acoustic and semantic information about sounds. Experimental results illustrate that audio embeddings encoding acoustic and semantic information achieve higher accuracy in the classification task. After careful analysis of classification errors, we identify some underlying reasons for failure and propose actions to mitigate them. The paper highlights the need for deeper exploration of all stages of classification, understanding the data and adopting methodologies capable of effectively handling data complexity and generalizing in real-world sound environments.

Index Terms**—** sound classification, sound taxonomies, machine learning, error characterization

1. INTRODUCTION

Sound classification plays a crucial role in numerous applications ranging from sound and music analysis, browsing and retrieval to acoustic monitoring and ubiquitous computing [1]. Automatic analysis of diverse sound types necessitates the extraction of relevant features from audio signals, combined with machine learning techniques. This has garnered significant attention from fields focused on music, speech, and environmental sounds, leading to the development of various taxonomies and algorithmic techniques tailored to different applications.

In this paper, we concentrate on a general-purpose classification framework where, instead of focusing on a particular type of sound, the goal is to classify *any* type of input sound. For that purpose, we previously developed the Broad Sound Taxonomy (BST), which organizes sounds into a two-level hierarchical structure with 5 top-level and 23 second-level classes [2]. The top level of the taxonomy consists of the classes *Music*, *Instrument samples*, *Speech*, *Sound effects*, and *Soundscapes*. A diagram with all classes (and their abbreviated names) can be seen in Fig. 1. The taxonomy is designed to be user-friendly and accommodates a wide diversity of sounds, ensuring the classes are easy to understand, broad, and comprehensive. These classes exhibit significant intra-class variability, primarily influenced by the semantic foundation upon which the taxonomy was constructed. Such intra-class variability means that sounds of the same class can exhibit very different acoustic characteristics. Our goal is to build a sound classification system that can successfully classify sounds using the BST taxonomy. To that end, we curate a dataset comprising 10k sounds annotated with the BST classes. We use k-NN classifiers and study their performance using input sound representations that capture different levels of acoustic and/or semantic information. Besides the classifier performance metrics, we conduct manual error analysis and systematically characterize the model's misclassifications. The moderate number of classes in the taxonomy proves advantageous in this step, enabling easier human evaluation of algorithmic mistakes. By analyzing misclassifications, we are able to suggest ways in which the classification system can be further improved.

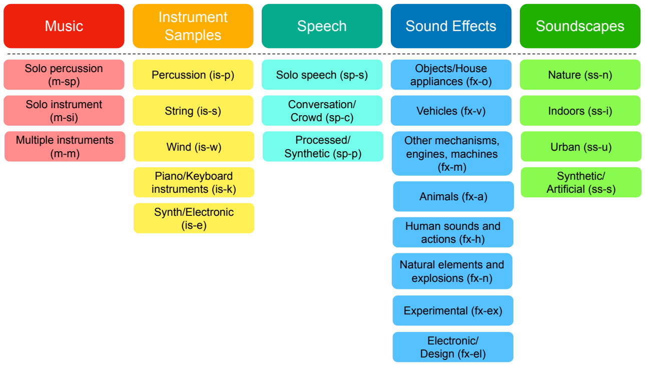

Music
Instrument Samples
Speech
Sound Effects
Soundscapes
Solo percussion (m-sp)
Solo instrument (m-si)
Multiple instruments (m-m)
Percussion (is-p)
String (is-s)
Wind (is-w)
Piano/Keyboard instruments (is-k)
Synth/Electronic (is-e)
Solo speech (sp-s)
Conversation/ Crowd (sp-c)
Processed/ Synthetic (sp-p)
Objects/House appliances (fx-o)
Vehicles (fx-v)
Other mechanisms, engines, machines (fx-m)
Animals (fx-a)
Human sounds and actions (fx-h)
Natural elements and explosions (fx-n)
Experimental (fx-ex)
Electronic/ Design (fx-el)
Nature (ss-n)
Indoors (ss-i)
Urban (ss-u)
Synthetic/ Artificial (ss-s)

Figure 1: Class hierarchy for the Broad Sound Taxonomy (BST).

The proposed approach and findings have broad applicability, as the automatic extraction of the systematized knowledge from such a hierarchical structure can streamline the organization, annotation, and retrieval of audio data, along with other related tasks across diverse domains. Using such a classifier, capable of categorizing any type of sound into broad categories, can be useful for providing an initial context of a sound class and thereafter for carrying out context-aware processing of sounds.

2. BACKGROUND

Over the years, several taxonomies have been proposed for organizing sound. Most taxonomies are tailored to specific domains or tasks, as exemplified by works on sound design [3, 4], urban or environmental scene analysis [5, 6, 7] and music or instrument categorization [8, 9, 10], while other taxonomies are designed to cover general use cases (e.g. Google's AudioSet [11]). On the one hand, when existing taxonomies are *simple* (i.e. low number of classes with shallow hierarchy), they tend to be domain-specific and are not comprehensive enough to generally classify *heterogeneous* sounds (e.g. ESC-50 [5], Urban Sound Taxonomy [6], FMA [8], NSynth [10]). On the other hand, general-purpose taxonomies are often very complex or lack a user-centric design (e.g. AudioSet has over 500 sound classes organized in a deep hierarchy), meaning that only expert users can use them effectively. The aforementioned Broad Sound Taxonomy addresses the lack of a simple yet comprehensive sound taxonomy that can be easily understood and used by sound practitioners of different levels of expertise and, at the same time, provide informative sound classes relevant to various applications such as sound analysis and retrieval [2].

In the field of machine listening, automatic sound classification has been typically addressed using machine-learning classifiers such as k-Nearest Neighbors (k-NNs), Support Vector Machines (SVMs), Multilayer Perceptrons (MLPs), and Hidden Markov Models (HMMs) [12]. These classifiers traditionally rely on features such as Mel-frequency cepstral coefficients (MFCCs) and other spectrum-based representations that only capture acoustic information of sounds. In recent years, different types of deep neural networks (DNNs) have gained prominence across the audio field due to their superior performance. One notable use is their ability to effectively transform raw audio data into highly meaningful representations. Because such representations are often obtained from models trained on classification tasks, they do not only capture acoustic information about sounds, but also encode some level of semantic information. Models such as VGGish[13], YAMNet [14], or FSD-SINet [15], produce high-level, semantically meaningful embeddings while using audio as input. Another recent approach is the use of contrastive learning techniques to train models that learn a joint audio and language embedding space in which sound semantics are even more prominent. An eminent example is the CLAP architecture [16, 17], which learns audio concepts from natural language sound descriptions. These learned feature representations can be used as input features with traditional machine learning classifiers for addressing downstream tasks, which is typically known as *transfer learning*. Through transfer learning, less complex models can efficiently leverage pre-trained models to achieve high accuracies in downstream tasks [18, 19, 20]. In this work, we use transfer learning to address the task of heterogeneous sound classification.

3. METHODOLOGY

3.1. Dataset creation

We introduce the Broad Sound Dataset (BSD), a collection of annotated sounds aligned with the second level of the classes defined in the BST taxonomy (Fig. 1). The initial release, a contribution of this paper, contains more than 10,000 sounds and is named BSD10k. BSD10k has been built using sounds obtained from Freesound, a website that hosts over 650,000 diverse sounds released under Creative Commons (CC) licenses and contributed by a wide range of individuals [21]. We leveraged existing public Freesound-based datasets such as FSD50K [22], freefield1010 [23], Freesound Loop Dataset [9], together with other in-house Freesound collections to compile an initial list of approximately 60,000 sound candidates of heterogeneous nature. These candidates were assigned to one of the five top-level classes of the BST taxonomy by leveraging their ground-truth labels from their original dataset(s) and using other heuristics based on basic signal processing techniques (e.g. onset detection) and available Freesound metadata (e.g. sound tags). After mapping the candidates to the top level of the taxonomy, a manual annotation phase was carried out to address potential inaccuracies and assign the corresponding second-level taxonomy category to each sound candidate.

For the annotation phase, we developed an in-house online annotation tool which was used by the authors of the paper to get familiar with the taxonomy and carry out the annotations. For each candidate sound, the annotators selected the most appropriate second-level class and provided a confidence level for their annotation. The provided confidence level is not used for the classification tasks in this paper, but it helps ensure a more accurate annotation process and may provide useful data for future experiments [24]. The original sound title and tags from Freesound were presented to the annotators to facilitate the annotation of acoustically ambiguous sounds. During the course of three months, the annotators classified 10,309 sounds, resulting in a total duration of 32.5 hours of audio, which forms the final BSD10k dataset. The annotated data has a non-uniform class distribution, leading to data imbalance, with some classes having over 1,000 sounds while others are represented by approximately 100 sounds. The top-level division of the audio data is 1635 *Music*, 2094 *Instrument samples*, 1250 *Speech*, 3911 *Sound effects*, and 1419 *Soundscapes*.

The Freesound audio data is heterogeneous, not only in content but also in quality, devices of recording, and lengths. Even though many sounds use (semi-)professional recording equipment [22], this diversity can be used as an advantage in developing a general-purpose classifier that generalizes well. During the annotation, we also monitored the diversity within each class; e.g. in the *Natural sounds and explosions* class, we ensured the presence of water sounds, rocks, as well as lightning and fireworks. The length of the sounds also varies, following a U-shape distribution. Longer samples were cropped to a maximum of 30 seconds, as sounds of this nature —often music or soundscapes— tend to repeat information. Even though we start with candidates from existing datasets, we download the original files using their IDs from the Freesound API. We then transform all sounds to adhere to a standardized format of uncompressed 44.1 kHz 16-bit mono audio files. The dataset is released with an open license and is publicly accessible[1].

3.2. Sound representations

We compare a selection of different types of sound representations, which are chosen to capture distinct levels of acoustic and semantic features.

FSSimRep: We extract a feature representation derived from various spectral, time-domain, rhythm and tonal characteristics calculated using signal-processing algorithms with the FreesoundExtractor[2] of the Essentia audio analysis library [25]. With an audio file given as input, the FreesoundExtractor provides several statistics for each of the features above, which are then aggregated into a vector of 846 dimensions and scaled to be in the range $[0, 1]$. The scaled vector is

[1] https://github.com/allholy/BSD10k

[2] https://essentia.upf.edu/freesound_extractor.html

reduced to 100 dimensions using Principal Component Analysis (PCA), producing the final sound representation. This representation is akin to the representation currently used for the sound similarity feature in Freesound, and it is expected to only capture the acoustic properties of sounds.

VGGish and FSD-SINet: We utilize the embeddings from VGGish [13] and FSD-SINet [15]. They are both large convolutional neural network (CNN) models trained on audio signals in classification tasks. These models take audio signals as input and are expected to learn both about their acoustic properties and semantic meaning by relating audio signals to the classification labels. We use both models as two examples of classification-based embeddings trained on distinct datasets (YouTube100M and FSD50K), with output representation dimensions of $(n, 128)$ and $(n, 512)$, respectively, where n represents the number of frames dependent on the length of the audio file. To obtain the final one-dimensional vector representation, we carry out temporal aggregation by averaging over n frames.

LAION-CLAP: Finally, in our experiments, we include embeddings extracted from the multi-modal LAION-CLAP model [17]. CLAP uses contrastive learning techniques to acquire knowledge from pairs of audio signals and natural language textual descriptions. This approach allows the model to be fed not only with the audio signals but also with rich contextual semantic information about them. Given an audio file as input, LAION-CLAP provides a final 512-dimensional vector representation.

3.3. Model and evaluation metrics

For our experimental setup, we use the k-Nearest Neighbors (k-NN) algorithm as our classifier. The choice is motivated by its low complexity, interpretability, and common use in transfer learning settings. To complement our experiments, we run preliminary experiments using Support Vector Machine (SVM) models and obtained results similar to those reported by k-NN models, therefore we will not report SVM results in this paper.

To identify the optimal hyperparameters, we compare various sets of model parameters to determine the most effective configuration for model performance. We conduct a grid search to systematically explore the hyperparameter space, evaluating different numbers of neighbors, distance metrics, and weighting schemes [26]. To evaluate the performance of the trained models, we calculate accuracy, precision, recall and F1-score evaluation metrics. We divide our dataset into two splits used for training and evaluation. The evaluation split consists of a random selection of 40 sounds for each second-level class of the taxonomy, totaling 920 sounds (9̃% of the size of the dataset). We assess qualitatively that the random selection for the evaluation set resulted in high intra-class sound variations. The rest of the sounds are included in the training set.

Additionally, we take advantage of the hierarchical structure of the taxonomy to run experiments using only the top-level classes as labels, grouping sounds with similar semantics and reducing the total number of classes to five (Fig. 1). For consistency, we use the same data split for the top-level training process. Although this approach introduces imbalance in the number of test samples per class due to the varying number of second-level classes within each top-level class, it ensures a fair comparison in the evaluation process.

To obtain further insights about classification performance, we characterize the errors by manually reviewing all misclassified sounds from the best-performing model across all input representations, as well as 200 randomly sampled misclassifications from the best models of the remaining input representations. This analysis is performed for both second-level and top-level classification setups. We identify the potential reasons for each misclassification and then consolidate the most common reasons into error categories.

Table 1: Accuracy and F1-score for the best-performing k-NN per input sound representation.

Model input	Second-level		Top-level	
	Accuracy	F1-score	Accuracy	F1-score
FSSimRep	0.426	0.40	0.678	0.667
VGGish	0.527	0.506	0.748	0.741
FSD-SINet	0.562	0.544	0.746	0.746
LAION-CLAP	**0.761**	**0.748**	**0.873**	**0.868**

4. RESULTS

4.1. Performance metrics

Table 1 shows the classification accuracies and F1-scores of the k-NN classifiers trained with the different input representations we compare. We report the accuracy and F1-score of the best-performing classifier for each input representation according to the hyperparameter optimization. We observe that, in almost all instances, the highest recall coincides with the highest accuracy. This suggests that comparing accuracies across various input representations, including the top-level classifiers with an unbalanced test set, remains a reliable metric without inherent bias towards classes with larger sample sizes.

Both accuracy and F1-score metrics show that classification performance improves when classifying at the top level compared to the second level of the taxonomy (average of 0.19 for accuracy and 0.21 for F1-score). This is expected as the task becomes progressively easier with fewer number of classes, introducing greater orthogonality at the first level of the taxonomy. The increase in performance comparing top-level with second-level is significantly lower for CLAP (0.11 for accuracy and 0.12 for F1-score), which could be attributed to the fact that CLAP captures sound semantics more efficiently and therefore it can perform better in the second-level of the taxonomy where class semantics are more nuanced.

The CLAP embeddings outperformed the other representations in both top-level and second-level classification tasks. This suggests that the joint audio-language embedding space captures acoustic and semantic information better, which is beneficial for the classification of heterogeneous sounds. VGGish and FSD-SINet result in very similar performances. Despite our expectation that FSD-SINet would outperform VGGish due to its training on the FSD50K dataset, which includes Freesound data relevant to our task, both models show comparable results. They have an average of 0.216 and 0.223 lower than CLAP for accuracy and F1-score, respectively. This suggests that these embeddings do not capture acoustic and semantic sound properties with the same richness as CLAP embeddings. Finally, the models trained with FSSimRep exhibit the lowest performance, an average of 0.101 and 0.106 lower than VGGish for accuracy and F1-score, respectively. This highlights the challenge of distinguishing classes using solely acoustic information due to intra-variability and acoustic diversity within classes.

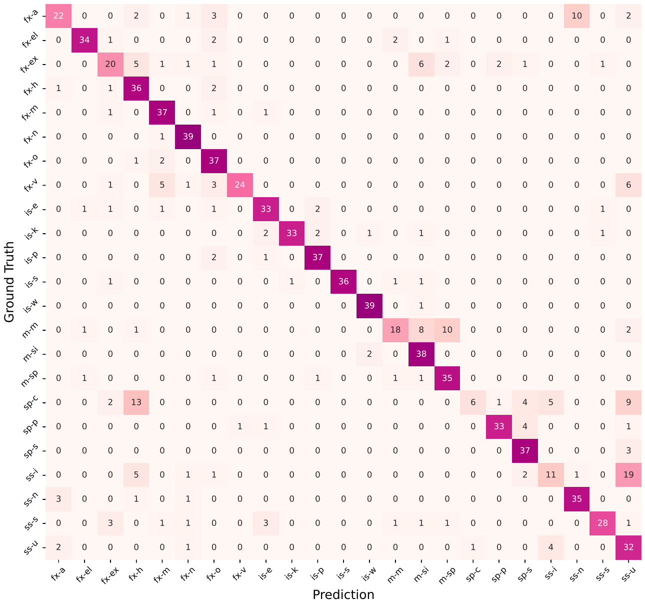


Figure 2: Confusion matrix for the best-performing k-NN model trained with CLAP.

Fig. 2 shows the confusion matrix for the best-performing k-NN model trained with the CLAP sound representation and holds insights into how the model performs for each individual second-level class of the taxonomy. We observe that most classes exhibit very good performance, yet there are instances of lower performance in specific classes, such as *Conversation/Crowd*. This discrepancy may stem from factors such as data imbalance, class complexity, or reduced orthogonality between certain classes.

Regarding hyperparameter optimization, we find that the variation in accuracy among the top 100 grid search configurations for each embedding training remains small, with a maximum difference of approximately 0.065 (and 0.035 for top classes). The top 100 choices include nearly all neighbors, distance metrics, and weighting schemes, indicating stable performance across a broad area of the hyperparameter space. This stability suggests that specific hyperparameters have little impact on the performance of this task when leveraging embeddings, regardless of their efficacy.

4.2. Error characterization

Table 2 shows the results of the error characterization. We observe that the most common reason for the misclassification of sounds is when sounds fall ambiguously *between classes*, either between second-level classes with a common top-level class (14.6%), or between second-level classes belonging to a different top-level class (26%). That suggests that even humans may have difficulty distinguishing these classes. Further insights about that matter could be obtained by analyzing the confidence annotation scores included in BSD10k. We also observe that simplifying the task in the top-level classification does not significantly reduce *between classes* errors. Interestingly, errors are more prevalent between different top-level categories than within the same one, indicating potential for enhancing the classifier's capability to differentiate between higher-level classes to improve overall hierarchical classification accuracy. Analyzing the discrepancies between the top-level and second-level classifiers reveals that 54% of errors across all second levels are accurately predicted by the top-level classifier, supporting the claim that integrating hierarchical information within a unified model is a promising future direction. Additionally, a notable portion of these errors are linked to the lowest-performing class (*Conversation/Crowd*), suggesting that improving the dataset or model to better handle less orthogonal classes could lead to better overall results.

Table 2: Error characterization for the best-performing k-NN model trained with CLAP.

Error category	**Second-level**	**Top-level**
Acoustic ambiguity	60	27
Between classes (different top)	57	52
Between classes (same top)	32	-
Common source	18	10
Prominence of one source	23	18
Single-source evolution	3	2
Low quality	3	0
Uncommon/Weird/Other	24	8
Total	**220**	**117**

Misclassifications due to *common source* (i.e classes include sounds from the same source), *single-source evolution* (i.e sound from one source evolves over time), or *prominence of one source* (i.e. one sound dominates in duration or loudness) are influenced by the taxonomy's nature, which separates sound samples even when they originate from the same source (e.g. birds as part of a soundscape *vs* isolated birds, or human talking *vs* human crying). Because of the class definitions, the model is tasked to learn deeper semantic distinctions and information about the source mixture, thereby making the classification task more complex. To reduce these errors, models could integrate mixture and context-aware learning strategies during training. Errors grouped under *acoustic ambiguity* have one or more acoustic properties that resemble another sound from a different class (sounds *like* x, *is* y). Emphasizing semantic information could mitigate these errors, as they are more pronounced in the lower-performing models with less semantic integration, constituting $43-54\%$ of their total errors (against $23-27\%$ for CLAP). We note, though, that confusing sounds with very high acoustic similarity may be less consequential in certain tasks, such as sound design.

5. CONCLUSIONS

In this paper, we present a comparative analysis of various input representations with different levels of acoustic and semantic information for the task of heterogeneous sound classification. To address the challenges posed by the classification of a broad taxonomy with significant intra-variability, we introduce the manually curated BSD10k dataset which enables automatic classification tasks and offers valuable data pools for diverse research tasks. To baseline the problem and understand the error margin, we complement the evaluation metrics with manual error characterization through auditory evaluation of the misclassifications. Our findings indicate that greater semantic information enhances classification performance and insertion of hierarchical information during training can prove beneficial. Organizing available data into simpler taxonomic structures can improve the sound description process and enable the training of reliable automatic classifiers, providing a pre-processing step for context-aware sound processing and understanding.

6. ACKNOWLEDGMENT

This research was partially funded by the Secretaria d'Universitats i Recerca del Departament de Recerca i Universitats de la Generalitat de Catalunya (ref. 2023FI-100252) under the Joan Oró program, and the IA y Música: Cátedra en Inteligencia Artificial y Música (TSI-100929-2023-1) by the Secretaría de Estado de Digitalización e Inteligencia Artificial and the European Union-Next Generation EU under the program Cátedras ENIA 2022.

7. REFERENCES

[1] F. Font, G. Roma, and X. Serra, "Sound sharing and retrieval," in *Computational Analysis of Sound Scenes and Events*. Springer, 2018, pp. 279–301.

[2] P. Anastasopoulou, X. Serra, and F. Font, "A General-Purpose Broad Taxonomy for the Classification of Heterogeneous Sound Collections," *Under review*, 2024.

[3] D. Moffat, D. Ronan, and J. D. Reiss, "Unsupervised taxonomy of sound effects," in *Proc. 20th Int. Conference on Digital Audio Effects (DAFx-17)*, 2017.

[4] T. Nielsen, J. Drury, and K. Paquin, "Universal Category System," https://universalcategorysystem.com/, 2020.

[5] K. J. Piczak, "ESC: Dataset for environmental sound classification," in *Proc. 23rd Int. Conference on Multimedia (ACM)*, 2015, pp. 1015–1018.

[6] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *Proc. 22nd Int. Conference on Multimedia (ACM)*, 2014, pp. 1041–1044.

[7] G. Lafay, M. Rossignol, N. Misdariis, M. Lagrange, and J.-F. Petiot, "Investigating the perception of soundscapes through acoustic scene simulation," *Behavior Research Methods*, vol. 51, pp. 532–555, 2019.

[8] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, "FMA: A dataset for music analysis," *arXiv preprint arXiv:1612.01840*, 2016.

[9] A. Ramires, F. Font, D. Bogdanov, J. B. Smith, Y.-H. Yang, J. Ching, B.-Y. Chen, Y.-K. Wu, H. Wei-Han, and X. Serra, "The Freesound loop dataset and annotation tool," in *Proc. 21st Int. Society for Music Information Retrieval (ISMIR)*, 2020.

[10] J. Engel, C. Resnick, A. Roberts, S. Dieleman, D. Eck, K. Simonyan, and M. Norouzi, "Neural audio synthesis of musical notes with WaveNet autoencoders," 2017.

[11] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. Int. Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 776–780.

[12] T. Virtanen, M. D. Plumbley, and D. Ellis, *Computational Analysis of Sound Scenes and Events*. Springer, 2018.

[13] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, *et al.*, "CNN architectures for large-scale audio classification," in *Proc. Int. Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 131–135.

[14] "Sound classification with YAMNet | TensorFlow Hub," https://www.tensorflow.org/hub/tutorials/yamnet.

[15] E. Fonseca, A. Ferraro, and X. Serra, "Improving sound event classification by increasing shift invariance in convolutional neural networks," in *arXiv Preprint arXiv:2107.00623*, 2021.

[16] B. Elizalde, S. Deshmukh, M. Al Ismail, and H. Wang, "CLAP learning audio concepts from natural language supervision," in *Proc. Int. Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023.

[17] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, "Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation," in *Proc. Int. Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023.

[18] P. Alonso-Jiménez, X. Serra, and D. Bogdanov, "Music representation learning based on editorial metadata from discogs," in *Proc. Int. Society of Music Information Retrieval (ISMIR)*, 2022, pp. 825–833.

[19] V. Sanguineti, P. Morerio, N. Pozzetti, D. Greco, M. Cristani, and V. Murino, "Leveraging acoustic images for effective self-supervised audio representation learning," in *Proc. 16th European Conference on Computer Vision (ECCV)*. Springer, 2020, pp. 119–135.

[20] D. Eck, P. Lamere, T. Bertin-mahieux, and S. Green, "Automatic generation of social tags for music recommendation," in *Advances in Neural Information Processing Systems*, J. Platt, D. Koller, Y. Singer, and S. Roweis, Eds., vol. 20. Curran Associates, Inc., 2007.

[21] F. Font, G. Roma, and X. Serra, "Freesound technical demo," in *Proc. 21st Int. Conference on Multimedia (ACM)*, 2013, pp. 411–412.

[22] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "FSD50K: An open dataset of human-labeled sound events," in *Proc. IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, 2021, pp. 829–852.

[23] D. Stowell and M. D. Plumbley, "An open dataset for research on audio field recording archives: Freefield1010," *arXiv preprint arXiv:1309.5275*, 2013.

[24] A. Mendez, M. Cartwright, J. Bello, and O. Nov, "Eliciting confidence for improving crowdsourced audio annotations," *Proc. on Human-Computer Interaction (ACM)*, vol. 6, 2022.

[25] D. Bogdanov, N. Wack, E. Gómez, S. Gulati, P. Herrera, O. Mayor, G. Roma, J. Salamon, J. Zapata, and X. Serra, "Essentia: An open-source library for sound and music analysis," in *Proc. 21st Int. Conference on Multimedia (ACM)*, 2013, pp. 855–858.

[26] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," *Neurocomputing*, vol. 415, pp. 295–316, 2020.

LEARNING MULTI-TARGET TDOA FEATURES FOR SOUND EVENT LOCALIZATION AND DETECTION

Axel Berg[1,2], *Johanna Engman*[1], *Jens Gulin*[1,3], *Karl Åström*[1], *Magnus Oskarsson*[1],

[1]Computer Vision and Machine Learning, Centre for Mathematical Sciences,
Lund University, Sweden {firstname.lastname@math.lth.se}
[2] Arm, Lund, Sweden {axel.berg@arm.com}
[3] Sony Europe B.V., Lund, Sweden {jens.gulin@sony.com}

ABSTRACT

Sound event localization and detection (SELD) systems using audio recordings from a microphone array rely on spatial cues for determining the location of sound events. As a consequence, the localization performance of such systems is to a large extent determined by the quality of the audio features that are used as inputs to the system. We propose a new feature, based on neural generalized cross-correlations with phase-transform (NGCC-PHAT), that learns audio representations suitable for localization. Using permutation invariant training for the time-difference of arrival (TDOA) estimation problem enables NGCC-PHAT to learn TDOA features for multiple overlapping sound events. These features can be used as a drop-in replacement for GCC-PHAT inputs to a SELD-network. We test our method on the STARSS23 dataset and demonstrate improved localization performance compared to using standard GCC-PHAT or SALSA-Lite input features.

Index Terms— sound event localization and detection, time difference of arrival, generalized cross-correlation

1. INTRODUCTION

The sound event localization and detection (SELD) task consists of classifying different types of acoustic events, while simultaneously localizing them in 3D space. The DCASE SELD Challenge [1] provides first order ambisonics (FOA) recordings and signals captured from a microphone array (MIC). In recent years, most systems submitted to the challenge have utilized the former format, whereas the latter has been less explored. In this work, we therefore focus on how to better exploit information in the MIC recordings by learning to extract better features.

Generalized cross-correlations with phase transform (GCC-PHAT) [2] combined with spectral audio features is the basis for most SELD methods for microphone arrays. The spectral features contain important cues on what type of sound event is active, whereas the purpose of GCC-PHAT is to extract the time-differences of arrival (TDOA) for pairs of microphones. The TDOA measurements can then be mapped to direction-of-arrival (DOA) estimates, given the geometry of the array. However, GCC-PHAT is known to be sensitive to noise and reverberation [3]. GCC-PHAT can also fail to separate TDOAs for overlapping events, since two events at different locations can have the same TDOA for a given microphone pair, which yields only one correlation peak.

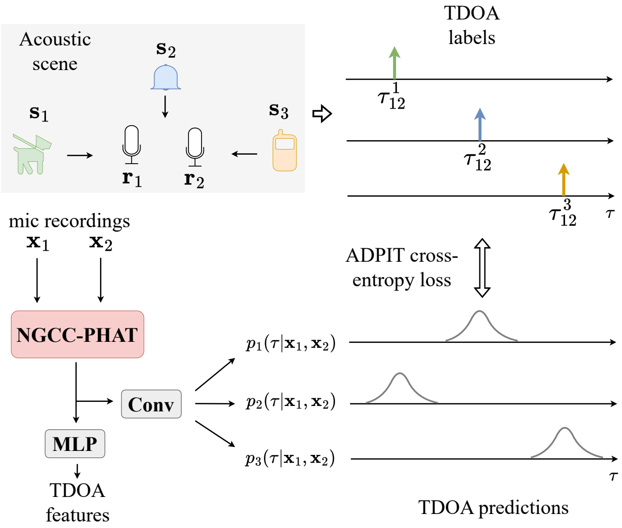


Figure 1: Overview of our pre-training strategy with $K = 3$ tracks. Given a set of sound events, we train a neural GCC-PHAT to predict the TDOA of each event. When the number of sound events is less than K, auxiliary duplication of the labels is used. In this illustration, only two microphones are shown for brevity.

To improve separation of overlapping events, Xu et al. [4] proposed a beamforming approach, where phase differences from the cross-power spectrum are used as input features. Similarly, Cheng et al. [5] showed that localization performance can be improved by first filtering the audio signals using a sound source separation network before performing feature extraction. Several works [6, 7] have also proposed end-to-end localization from raw audio signals. The most widely adopted input feature is however the spatial cue-augmented log-spectrogram (SALSA) [8] and variants thereof (SALSA-Lite) [9], that combine directional cues with spectral cues in a single feature. This is done by calculating the principal eigenvector of the spatial covariance matrix for the different frequencies in the spectrogram.

Although some recent works [10, 11, 12] have approached TDOA estimation using learning-based methods, there is a lack

This work was partially supported by the strategic research project ELLIIT and the Wallenberg AI, Autonomous Systems and Software Program (WASP), funded by the Knut and Alice Wallenberg (KAW) Foundation. Model training was enabled by the Berzelius resource provided by the KAW Foundation at the National Supercomputer Centre in Sweden.

Code: `https://github.com/axeber01/ngcc-seld/`

of research in how to combine this with the SELD task. Berg et al. [12] proposed using a shift-equivariant neural GCC-PHAT (NGCC-PHAT) network. However, this method, as it was originally proposed, only supports single-source TDOA estimation and was not evaluated in a real-world localization scenario.

In this work, we describe how NGCC-PHAT can be trained to extract TDOA features for multiple sound sources. We show that such features can be learnt by employing permutation invariant training, which allows for prediction of TDOAs for multiple overlapping sound events. Furthermore, we show that these features can be used with an existing SELD-pipeline on a real-world dataset, for better performance compared to using traditional input features. The material presented in this work is an extension of our DCASE 2024 challenge submission [13].

2. METHOD

2.1. Background

Consider an acoustic scene, as shown in Figure 1, with M microphones located at positions $\mathbf{r}_m \in \mathbb{R}^3$ for $m = 1, \dots, M$. Furthermore, let $\mathbf{s}_p \in \mathbb{R}^3$, $p = 1, \dots, P$ denote the locations of the active sound events. For a given time frame, each microphone records a signal x_i, which is composed of the sum of active events as

$$x_i[n] = \sum_{p=1}^{P} (h_{p,i} * u_p)[n] + w_i[n], \quad n = 1, \dots, N, \tag{1}$$

where u_p is the p:th active event, $h_{p,i}$ is the room impulse response from the p:th event to the i:th microphone, w_i is additive noise and N is the number of samples. Furthermore, we define the TDOA for microphone pair (i, j) and the p:th event as

$$\tau_{ij}^p = \lfloor \frac{F_s}{c} \left(||\mathbf{s}_p - \mathbf{r}_i||_2 - ||\mathbf{s}_p - \mathbf{r}_j||_2 \right) \rceil, \tag{2}$$

where F_s is the sampling rate, c is the speed of sound and $\lfloor \cdot \rceil$ denotes rounding to the nearest integer.

The GCC-PHAT is defined as

$$R_{ij}[\tau] = \frac{1}{N} \sum_{k=0}^{N-1} \frac{X_i[k] X_j^*[k]}{|X_i[k] X_j^*[k]|} e^{\frac{i 2\pi k \tau}{N}}, \tag{3}$$

where (X_i, X_j) are the discrete Fourier transforms of (x_i, x_j). The feature is calculated for time delays $\tau = -\tau_{\text{max}}, \dots, \tau_{\text{max}}$, where $\tau_{\text{max}} = \max_{i,j} \lfloor ||\mathbf{r}_i - \mathbf{r}_j||_2 F_s / c \rceil$ is the largest possible TDOA for any pair of microphones. In an anechoic and noise-free environment with a single sound event u_p, this results in $R_{ij}[\tau] = \delta_{\tau_{ij}^p}[\tau]$, where

$$\delta_{\tau_{ij}^p}[\tau] = \begin{cases} 1, & \tau = \tau_{ij}^p, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

In practice, GCC-PHAT will often yield incorrect TDOA estimates due to noise and reverberation. In the case of multiple overlapping sound events, the different events may interfere and result in difficulties resolving peaks in their signal correlations.

NGCC-PHAT attempts to alleviate this problem by filtering the input signals using a learnable filter bank with L convolutional filters, before computing GCC-PHAT features R_{ij}^l, $l = 1, \dots, L$ for each channel in the signals independently. In theory, such a filter bank can perform source separation so that different channels in the NGCC-PHAT correspond to TDOAs for different sound events. Note that for an ideal filter bank that perfectly separates the p:th sound event to the l:th channel, we would have $R_{ij}^l[\tau] = \delta_{\tau_{ij}^p}[\tau]$ in an anechoic and noise-free environment, due to the shift-equivariance of the convolutional filters.

2.2. Permutation Invariant Training for TDOA Estimation

We extend NGCC-PHAT to predict time delays for multiple events in a single time frame using auxiliary duplicating permutation invariant training (ADPIT) [14], by creating separate target labels for each active sound event. This is done by training a classifier network to predict the TDOA of all active events for all pairs of microphones by treating it as a multinomial classification problem. The L correlation features are first processed using another series of convolutional layers with C output channels. These are then projected to K different output tracks which are assigned to the different events. The last layer of the NGCC-PHAT network therefore outputs probability distributions $p_k(\tau | \mathbf{x}_i, \mathbf{x}_j)$ for $k = 1, \dots, K$ over the set of integer delays $\tau \in \{-\tau_{\text{max}}, \dots, \tau_{\text{max}}\}$, as illustrated in Figure 1.

With K as the number of tracks, assume for now that there are also $P = K$ active events. Furthermore, let $\text{Perm}([K])$ denote the set of permutations of the events $\{1, \dots, K\}$. For a single microphone pair (i, j) and an event arrangement $\alpha \in \text{Perm}([K])$, the loss is calculated using the average cross-entropy over all output tracks as

$$l_\alpha(\mathbf{x}_i, \mathbf{x}_j) = -\frac{1}{K} \sum_{k=1}^{K} \sum_{\tau=-\tau_{\text{max}}}^{\tau_{\text{max}}} \delta_{\tau_{ij}^{\alpha(k)}}[\tau] \log p_k(\tau | \mathbf{x}_i, \mathbf{x}_j). \tag{5}$$

Due to the ambiguity in assigning different output tracks to different events, we calculate the loss for all possible permutations of the events and use the minimum. The loss is then averaged over all $M(M-1)/2$ microphone pairs, giving the total loss

$$\mathcal{L} = \frac{2}{M(M-1)} \sum_{\substack{i,j=1 \\ i<j}}^{M} \min_{\alpha \in \text{Perm}([K])} l_\alpha(\mathbf{x}_i, \mathbf{x}_j). \tag{6}$$

Note that this loss function is class-agnostic, since the output tracks are not assigned class-wise. The main purpose of the TDOA features are therefore to provide better features for localization when combined with spectral features that are suitable for classification.

When the assumption $P = K$ does not hold, the formal implication is that α needs to cover another set of event arrangements. Our approach is equivalently to transform each such case into subcases where the assumption holds. Time frames with no active events ($P = 0$) are discarded in the loss calculation, since no TDOA label can be assigned. When $1 \leq P \leq K - 1$, we perform auxiliary duplication of events following the method in [14], which makes the loss invariant to both permutations and which events that are duplicated. Furthermore, in the case of $K < P$, it is possible to compute the loss for all subsets of K events from P and use the minimum.

3. EXPERIMENTAL SETUP

3.1. Using TDOA Features for SELD

In order to show the benefits of better TDOA features for SELD, we demonstrate how they can be used in conjunction with a SELD-system. This involves two training phases: 1) pre-training of the

NGCC-PHAT network for TDOA prediction and 2) training the SELD-network using the TDOA features as input. The NGCC-PHAT network operates on raw audio signals and consists of four convolutional layers, the first being a SincNet [15] layer, and the remaining three use filters of length 11, 9, and 7 respectively. Here, each convolutional layer has $L = 32$ channels and together form the filter bank mentioned in Section 2.1, which is applied independently to audio from the different microphones. GCC-PHAT features are computed channel-wise for each microphone pair, and the features are then processed by another four convolutional layers, where the final layer has $C = 16$ output channels.

The maximum delay used is chosen for compatibility with the setup in the STARSS23 dataset [16], which uses a tetrahedral array with $M = 4$ microphones. The diameter of the array is 8.4 cm, which corresponds to a maximum TDOA of $\tau_{\text{max}} = 6$ delays at a sampling rate of $F_s = 24$ kHz. In total, the TDOA features therefore have shape $[C, M(M-1)/2, 2\tau_{\text{max}} + 1] = [16, 6, 13]$.

During pre-training for TDOA-prediction, the 16 channels are then mapped by a convolutional layer to $K = 3$ output tracks. Although the maximum polyphony in a single time frame in the dataset is five, we use $K = 3$ tracks since the computational complexity of permutation invariant training scales as $\mathcal{O}(K!)$ and more than three simultaneous events are rare. When more than three events are active, for pre-training we randomly select labels for three events and discard the rest.

When training the SELD-network, we extract the TDOA input features for longer audio signals by windowing the NGCC-PHAT computation without overlap. We use an input duration of 5 second audio inputs, which corresponds to $T = 250$ TDOA features when using a window length of 20 ms. Since the TDOA features are designed to be class-agnostic, we combine them with spectral features for the same time-frame in order to better distinguish between different types of event. For this we use log mel-spectograms (MS) with $F = 64$ spectral features for each recording.

When merging the spectral features with the TDOA features, we first concatenate the 16 channels for the 6 microphone pairs of the TDOA features, and use a multi-layer perceptron to map the 13 time-delays to 64 dimensions. The TDOA features are then reshaped and concatenated with the M spectral features channel-wise, as shown in Figure 2, resulting in a combined feature size of $[CM(M-1)/2 + M, T, F] = [100, 250, 64]$.

The combined feature is passed through a small convolutional network with 64 output channels with pooling over the time and spectral dimensions. Here we use two pooling variants that determine the size of the input features to the SELD-network: 1) pooling over 5 time windows and 4 frequencies, which produces features of size [64, 50, 16], or 2) pooling over 5 time windows and no pooling over frequencies, which results in features of size [64, 50, 64]. We call the resulting network variants *Small* and *Large* for this reason.

For SELD-training, we use a CST-Former [17] network that consists of Transformer blocks, where each block contains three self-attention modules: temporal attention, spectral attention and channel attention with unfolded local embedding. We use the default configuration with two blocks, each with eight attention heads, and refer to [17] for more details about this architecture.

3.2. Dataset and Model Training

We train all our models on a mixture of real spatial audio recordings and simulated recordings. The real recordings are from the STARSS23 [16] audio-only dev-train dataset, which consists of about 3.5 hours of multi-channel audio recordings. The dataset has up to 5 simultaneous events from 13 different classes. For data augmentation, we use channel-swapping [18], which expands the dataset by a factor of 8 by swapping the input channels and corresponding DOA labels in different combinations.

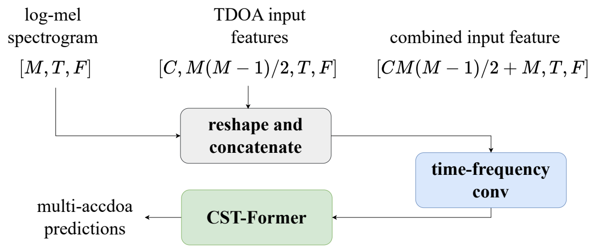


Figure 2: Illustration of how TDOA features are used together with log mel-spectrograms as input to the CST-Former network.

The simulated data is provided as a part of the DCASE 2024 challenge [19] and consists of 20 hours of synthesized recordings, where the audio is taken from the FSD50K [20] dataset. In addition, we generate an additional 2 hours of synthesized recordings using Spatial Scaper [21] with impulse responses from the TAU [22] and METU [23] databases. This additional data contains sounds from classes that occur rarely in STARSS23, namely “bell”, “clapping”, “doorCupboard”, “footsteps”, “knock” and “telephone”. The total amount of training data is about 50 hours.

The NGCC-PHAT network was trained for one epoch with a constant learning rate of 0.001, after which the weights were frozen. The CST-Former network was then trained for 300 epochs using the AdamW optimizer [24] with a batch size of 64, a cosine learning rate schedule starting at 0.001 and weight decay of 0.05. The mean squared error was used as loss function with labels in the Multi-ACCDOA [14] format, with distances included as proposed in [25]. In order to penalize errors in predicted distance relative to the proximity of the sound events, we scale loss-terms for the distance error with the reciprocal of the ground truth distance.

Evaluations were done using the DCASE 2024 SELD challenge metrics [1, 26]. This includes the location dependent F-score F_{LD}, the DOA error $DOAE$ and the relative distance error RDE, which is the distance error divided by the ground truth distance to the event. Each metric is calculated class-wise and then macro-averaged across all classes. Furthermore, the location dependent F-score only counts predicted events as true positives if they are correctly classified and localized, such that predictions with $DOAE$ larger than $T_{DOA} = 20^\circ$ or RDE larger than $T_{RD} = 1$ are counted as false positives. We focus on evaluating the performance of our method compared to that of other commonly used input features with the same SELD-network, and do not compare to other (e.g. FOA-based) state-of-the-art methods.

4. RESULTS

Our main results are presented in Table 1, where we compare our method to GCC with MS and to SALSA-Lite. Our method performs better in terms of F_{LD} and $DOAE$, for both the Small and Large variant of the network, although SALSA-Lite has the lowest RDE for the Large variant. When increasing the model size, the results improve for both SALSA-Lite and NGCC, but not for GCC. Since GCC features are less informative, the increase in model size results

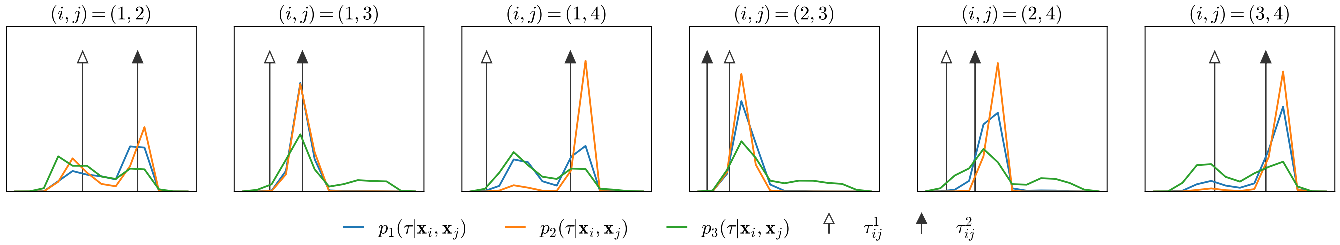


Figure 3: An example of the TDOA predictions $p_k(\tau|\mathbf{x}_i, \mathbf{x}_j)$ from the pre-trained NGCC-PHAT network using $K = 3$ output tracks. Predictions are shown for all six microphone combinations (i, j) at a single time frame with two events and ground truth TDOAs τ_{ij}^1 and τ_{ij}^2.

Table 1: Macro-averaged test results on STARSS23 [16] dev-test.

Input feature	F_{LD} ↑	$DOAE$ ↓	RDE ↓	#params
CST-Former Small				
GCC + MS	15.7 ± 1.0	27.7 ± 2.1	0.78 ± 0.02	550K
SALSA-Lite	24.6 ± 2.0	27.0 ± 1.2	$\mathbf{0.41 \pm 0.02}$	530K
NGCC + MS	$\mathbf{26.0 \pm 2.0}$	$\mathbf{25.8 \pm 2.3}$	0.42 ± 0.01	663K
CST-Former Large				
GCC + MS	14.2 ± 1.1	28.4 ± 1.9	0.84 ± 0.03	1.37M
SALSA-Lite	26.1 ± 1.0	26.4 ± 3.6	$\mathbf{0.42 \pm 0.02}$	1.35M
NGCC + MS	$\mathbf{28.2 \pm 2.8}$	$\mathbf{23.2 \pm 1.8}$	0.50 ± 0.02	1.49M

Table 2: Ablations of the number input channels used in the TDOA input features for CST-Former Small.

C	F_{LD} ↑	$DOAE$ ↓	RDE ↓	#params
1	24.4 ± 2.3	29.7 ± 3.3	0.44 ± 0.08	608K
4	24.2 ± 0.8	$\mathbf{23.2 \pm 2.5}$	0.46 ± 0.01	619K
16	$\mathbf{26.0 \pm 2.0}$	25.8 ± 2.3	$\mathbf{0.42 \pm 0.01}$	663K

in overfitting. The same can be said for the increase in RDE when using NGCC + MS, since the TDOA features from both GCC and NGCC mostly contain angular cues, but less information about spatial distance. Note that GCC + MS and NGCC + MS use exactly the same CST-Former architecture, so the extra parameter count when using NGCC comes from the pre-trained feature extractor. When using SALSA-Lite, the pooling operations in the convolutional layers were adjusted in order to achieve a similar model size.

In order to verify the importance of using more than one input channel for TDOA features, we ablate the number of channels C in the NGCC-PHAT network. The results are shown in Table 2, where it can be seen that increasing the number of channels from 1 to 16 increases performance in terms of all metrics. This agrees with the intuition that using more than one input channels enables the pre-training to better separate spatial cues from different events. Furthermore, the cost for increasing the number channels in terms of the increase in model parameters is relatively small.

We also ablate the number of tracks K used for TDOA-prediction during pre-training, and present the location dependent F-score for values of T_{DOA} in Figure 4. Due to the sensitivity of the macro-averaged F-score to incorrect predictions for rare classes in the test data, we instead use the micro-averaged statistic. At the default 20° threshold, the effect of increasing the number of tracks is small, but asymptotically it is clear that using $K = 3$ tracks increases the F-score regardless of how many events are active. Note that the number of tracks only affects the complexity in the pre-training stage of NGCC-PHAT, and not the overall parameter count of the final model, since all C channels are used as input to the network, and the mapping to K tracks can be discarded.

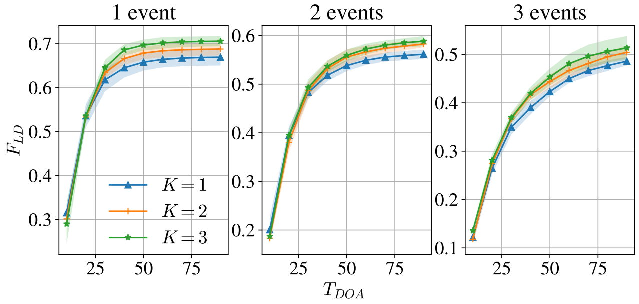


Figure 4: Micro-averaged F-score as a function of the angular threshold T_{DOA} using different number of output tracks K during TDOA pre-training. Evaluation was done using CST-Former Small.

Finally, we show examples of TDOA predictions in Figure 3. When the TDOAs of the events are well-separated, the different tracks yield different peaks at approximately the correct time delays. However, for the microphone pairs where events are tightly spaced, the predictions fail to separate the different TDOAs.

5. CONCLUSIONS

In this work we proposed an input feature based on NGCC-PHAT and showed its usefulness as input to a SELD-network. Permutation invariant training for the TDOA estimation problem enabled NGCC-PHAT to learn TDOA features for multiple overlapping sound events, and improved SELD performance compared to using GCC-PHAT or SALSA-Lite input features.

These results indicate that our NGCC-PHAT pre-training for TDOA classification provides a good feature extractor for the SELD task. Intuitively, better TDOA prediction in the feature extractor ought to yield better SELD results, but further studies are needed to validate this. Evaluating TDOA prediction performance would however involve new methodology, such as heuristics for peak selection from the output tracks, as well as selecting useful evaluation metrics. The downstream network could be resilient to some type of information our current loss function aims to suppress. In addition, a source-wise or class-wise TDOA format could be beneficial. We therefore anticipate future work to explore other pre-training options and end-to-end training.

Focusing on the feature extractor, we made minimal effort to address the other challenges of the dataset. We leave for future work to incorporate known techniques, such as class balancing, additional data augmentation, temporal filtering and ensemble voting.

6. REFERENCES

[1] "Audio and Audiovisual Sound Event Localization and Detection with Source Distance Estimation," https://dcase.community/challenge2024/task-audio-and-audiovisual-sound-event-localization-and-detection-with-source-distance-estimation, 2024, [Accessed 2024-07-03].

[2] C. Knapp and G. Carter, "The Generalized Correlation Method for Estimation of Time Delay," *IEEE transactions on acoustics, speech, and signal processing*, vol. 24, no. 4, pp. 320–327, 1976.

[3] B. Champagne, S. Bédard, and A. Stéphenne, "Performance of time-delay estimation in the presence of room reverberation," *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 2, pp. 148–152, 1996.

[4] W. Xue, Y. Tong, C. Zhang, G. Ding, X. He, and B. Zhou, "Sound Event Localization and Detection Based on Multiple DOA Beamforming and Multi-Task Learning," in *Proc. Interspeech 2020*, 2020, pp. 5091–5095.

[5] S. Cheng, J. Du, Q. Wang, Y. Jiang, Z. Nian, S. Niu, C.-H. Lee, Y. Gao, and W. Zhang, "Improving sound event localization and detection with class-dependent sound separation for real-world scenarios," in *2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2023, pp. 2068–2073.

[6] H. Sundar, W. Wang, M. Sun, and C. Wang, "Raw waveform based end-to-end deep convolutional network for spatial localization of multiple acoustic sources," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 4642–4646.

[7] Y. He and A. Markham, "SoundDoA: Learn Sound Source Direction of Arrival and Semantics from Sound Raw Waveforms," in *Proc. Interspeech 2022*, 2022, pp. 2408–2412.

[8] T. N. T. Nguyen, K. N. Watcharasupat, N. K. Nguyen, D. L. Jones, and W.-S. Gan, "Salsa: Spatial cue-augmented log-spectrogram features for polyphonic sound event localization and detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 1749–1762, 2022.

[9] T. N. T. Nguyen, D. L. Jones, K. N. Watcharasupat, H. Phan, and W.-S. Gan, "SALSA-Lite: A Fast and Effective Feature for Polyphonic Sound Event Localization and Detection with Microphone Arrays," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 716–720.

[10] X. Xiao, S. Zhao, X. Zhong, D. L. Jones, E. S. Chng, and H. Li, "A learning-based approach to direction of arrival estimation in noisy and reverberant environments," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 2814–2818.

[11] D. Salvati, C. Drioli, and G. L. Foresti, "Time Delay Estimation for Speaker Localization Using CNN-Based Parametrized GCC-PHAT Features," in *Interspeech*, 2021, pp. 1479–1483.

[12] A. Berg, M. O'Connor, K. Åström, and M. Oskarsson, "Extending GCC-PHAT using Shift Equivariant Neural Networks," in *Proc. Interspeech 2022*, 2022, pp. 1791–1795.

[13] A. Berg, J. Engman, J. Gulin, K. Åström, and M. Oskarsson, "The LU System for DCASE 2024 Sound Event Localization and Detection Challenge," DCASE2024 Challenge, Tech. Rep., June 2024.

[14] K. Shimada, Y. Koyama, S. Takahashi, N. Takahashi, E. Tsunoo, and Y. Mitsufuji, "Multi-ACCDOA: Localizing and detecting overlapping sounds from the same class with auxiliary duplicating permutation invariant training," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 316–320.

[15] M. Ravanelli and Y. Bengio, "Speaker recognition from raw waveform with sincnet," in *2018 IEEE spoken language technology workshop (SLT)*. IEEE, 2018, pp. 1021–1028.

[16] K. Shimada, A. Politis, P. Sudarsanam, D. A. Krause, K. Uchida, S. Adavanne, A. Hakala, Y. Koyama, N. Takahashi, S. Takahashi, T. Virtanen, and Y. Mitsufuji, "STARSS23: An audio-visual dataset of spatial recordings of real scenes with spatiotemporal annotations of sound events," in *Advances in Neural Information Processing Systems*, vol. 36. Curran Associates, Inc., 2023, pp. 72 931–72 957.

[17] Y. Shul and J.-W. Choi, "CST-Former: Transformer with channel-spectro-temporal attention for sound event localization and detection," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 8686–8690.

[18] Q. Wang, J. Du, H.-X. Wu, J. Pan, F. Ma, and C.-H. Lee, "A four-stage data augmentation approach to resnet-conformer based acoustic modeling for sound event localization and detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 1251–1264, 2023.

[19] D. A. Krause and A. Politis, "[DCASE2024 Task 3] Synthetic SELD mixtures for baseline training," Apr. 2024. [Online]. Available: https://doi.org/10.5281/zenodo.10932241

[20] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "FSD50K: An open dataset of human-labeled sound events," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 829–852, 2021.

[21] I. R. Roman, C. Ick, S. Ding, A. S. Roman, B. McFee, and J. P. Bello, "Spatial Scaper: a library to simulate and augment soundscapes for sound event localization and detection in realistic rooms," *arXiv preprint arXiv:2401.12238*, 2024.

[22] A. Politis, S. Adavanne, and T. Virtanen, "TAU Spatial Room Impulse Response Database (TAU-SRIR DB)," Apr. 2022. [Online]. Available: https://doi.org/10.5281/zenodo.6408611

[23] O. Olgun and H. Hacihabiboglu, "METU SPARG Eigenmike em32 Acoustic Impulse Response Dataset v0.1.0," Apr. 2019. [Online]. Available: https://doi.org/10.5281/zenodo.2635758

[24] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[25] D. A. Krause, A. Politis, and A. Mesaros, "Sound event detection and localization with distance estimation," *arXiv preprint arXiv:2403.11827*, 2024.

[26] A. Politis, A. Mesaros, S. Adavanne, T. Heittola, and T. Virtanen, "Overview and evaluation of sound event localization and detection in dcase 2019," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 684–698, 2020.

LEVERAGING SELF-SUPERVISED AUDIO REPRESENTATIONS FOR DATA-EFFICIENT ACOUSTIC SCENE CLASSIFICATION

Yiqiang Cai[1], *Shengchen Li*[1], *Xi Shao*[2]

[1] Xi'an Jiaotong-Liverpool University, School of Advanced Technology, Suzhou, China,
yiqiang.cai21@student.xjtlu.edu.cn, shengchen.li@xjtlu.edu.cn
[2] Nanjing University of Posts and Telecommunications,
College of Telecommunications and Information Engineering, Nanjing, China,
shaoxi@njupt.edu.cn

ABSTRACT

Acoustic scene classification (ASC) predominantly relies on supervised approaches. However, acquiring labeled data for training ASC models is often costly and time-consuming. Recently, self-supervised learning (SSL) has emerged as a powerful method for extracting features from unlabeled audio data, benefiting many downstream audio tasks. This paper proposes a data-efficient and low-complexity ASC system by leveraging self-supervised audio representations extracted from general-purpose audio datasets. We introduce BEATs, an audio SSL pre-trained model, to extract the general representations from AudioSet. Through extensive experiments, it has been demonstrated that the self-supervised audio representations can help to achieve high ASC accuracy with limited labeled fine-tuning data. Furthermore, we find that ensembling the SSL models fine-tuned with different strategies contributes to a further performance improvement. To meet low-complexity requirements, we use knowledge distillation to transfer the self-supervised knowledge from large teacher models to an efficient student model. The experimental results suggest that the self-supervised teachers effectively improve the classification accuracy of the student model. Our best-performing system obtains an average accuracy of 56.7%[1].

Index Terms— Acoustic scene classification, data efficiency, self-supervised learning, fine-tuning, knowledge distillation

1. INTRODUCTION

Acoustic Scene Classification (ASC) is a task to recognize the environment in which an audio recording was captured, such as streets, parks, or airports [1]. Traditional approaches to ASC typically rely on supervised learning techniques [2, 3, 4, 5], which require large, labeled datasets to perform effectively. However, obtaining such labeled datasets is a resource-intensive process, often involving extensive manual annotation and data collection efforts. In the task 1 of the DCASE 2024 Challenge, participants are required to create low-complexity ASC systems that are trained with limited labeled data [6]. Specifically, five training subsets are provided, including 5%, 10%, 25%, 50%, and 100% of the original training set's size. The performance of the submitted systems, trained on 5 subsets, is assessed by the average accuracy. This task encourages the development of efficient models capable of maintaining high performance despite reduced training data, advancing the practical applicability and scalability of ASC systems in real world.

In recent years, self-supervised learning (SSL) has been widely applied to address the scarcity of labeled data in audio tasks. SSL leverages the structure of the data to create supervisory signals, allowing models to learn meaningful representations from unlabeled audio data. SSAST [7] introduces a masking strategy on the input spectrogram patches, allowing the transformer model to be pre-trained using both reconstruction loss and contrastive loss. Similarly, Audio-MAE [8] and MaskSpec [9] pre-train an encoder-decoder transformer architecture by reconstructing the original audio spectrogram from its masked version. BEATs [10] focuses on pre-training the transformer encoder by predicting the discrete labels generated by an acoustic tokenizer. After SSL pre-training on the general-purpose datasets, these models can be fine-tuned for various labeled tasks, such as keyword spotting and sound event detection. However, the application of audio self-supervised pre-trained models to ASC has been relatively unexplored.

In this work, we propose a data-efficient and low-complexity system with audio self-supervised pre-trained models for ASC. In Section 2, BEATs [10], an audio transformer model SSL pre-trained on AudioSet [11], is introduced. The pre-trained encoders of models are then appended with a new linear classifier and fine-tuned on the ASC dataset. We experiment with various fine-tuning strategies and data augmentation techniques. The results demonstrate that the self-supervised representations extracted from the general-purpose audio dataset can significantly improve the ASC accuracy with limited labeled data. Moreover, it has been found that the ensemble of SSL models fine-tuned with different strategies makes further improvements to the ASC performance. Section 3 focuses on addressing the complexity requirements, where a knowledge distillation framework [3] is used to transfer the self-supervised knowledge from BEATs to TF-SepNet-64 [12, 13], which is an efficient CNN-based ASC model. The experimental results and ablation study are detailed in Section 4. It shows that the self-supervised teachers significantly improve the performance of student model, achieving an average accuracy of 56.7%. Our submitted system ranked 4th in the DCASE 2024 Challenge [13].

2. SELF-SUPERVISED PRE-TRAINING AND FINE-TUNING

In this section, we aim to achieve high ASC accuracy with limited labeled data by leveraging the self-supervised audio representations. Specifically, we introduce BEATs, a state-of-the-art audio SSL model, to extract the general features from AudioSet [11]. The

[1] `https://github.com/yqcai888/easy_dcase_task1`

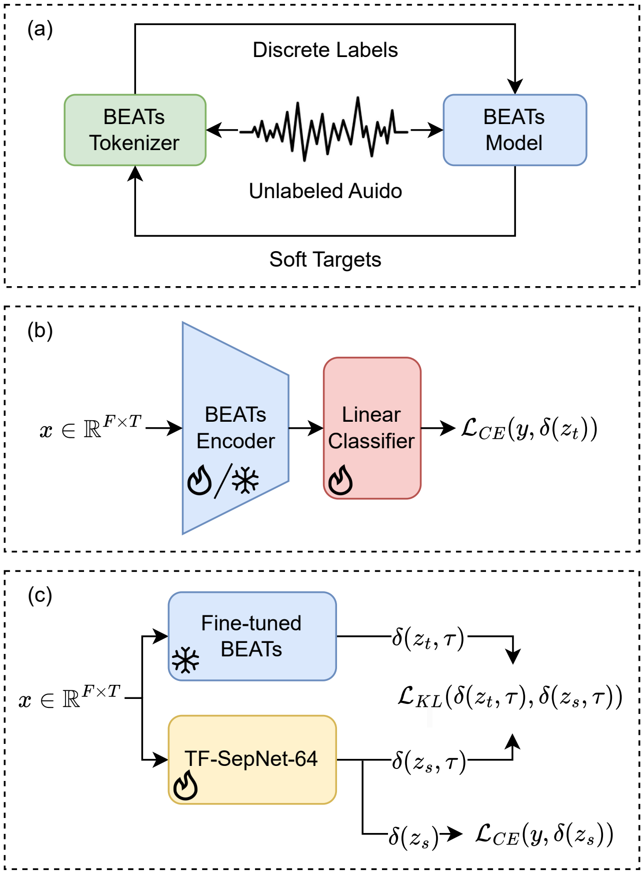


Figure 1: Proposed data-efficient and low-complexity ASC system. (a) Self-supervised pre-training BEATs on AudioSet. (b) Fine-tuning pre-trained BEATs on ASC dataset. (c) Distilling knowledge from fine-tuned BEATs to TF-SepNet-64. **Snowflake** icon indicates that the parameters of the corresponding part are frozen, while **Flame** icon indicates the opposite.

SSL pre-trained models are then experimented with two fine-tuning strategies, frozen fine-tuning and unfrozen fine-tuning, for adapting to the ASC task. Experimental results are presented in Section 4.1.

2.1. BEATs

Bidirectional Encoder representation from Audio Transformers (BEATs) [10] is an audio pre-training framework that iteratively optimizes an acoustic tokenizer and an audio SSL model. As illustrated in Figure 1 (a), the BEATs tokenizer generates discrete labels of unlabeled audio, which the BEATs model learns to predict. Concurrently, the tokenizer is trained by distilling knowledge from the pre-trained BEATs model, enabling iterative optimization of both components. The authors argue that discrete label prediction captures high-level audio semantics more effectively than the reconstruction loss used in previous audio SSL models. The tokenizer and label predictor of SSL model are discarded after self-supervised pre-training. In the original work, BEATs models are self-supervised pre-trained, and alternatively supervised fine-tuned, on AudioSet before applying to downstream tasks. For convinience, we denote the purely self-supervised pre-trained BEATs model as BEATs (SSL). The SSL pre-trained BEATs with additional supervised fine-tuning on AudioSet is denoted as BEATs (SSL+SL).

Model	5%	10%	25%	50%	100%	Avg.
BEATs (SSL)*	50.7	52.0	54.2	55.0	55.8	53.5
BEATs (SSL)	52.9	54.9	58.1	59.7	61.2	57.4
BEATs (SSL+SL)	54.3	56.6	59.7	60.7	62.1	58.7
3 Ensemble	55.4	57.6	61.1	62.2	64.2	60.1
12 Ensemble	**55.8**	**58.0**	**61.6**	**62.9**	**64.6**	**60.6**

Table 1: Accuracy of fine-tuned BEATs on the test set of TAU Urban Acoustic Scene 2022 Mobile development dataset [14]. **SSL** denotes the BEATs model is self-supervised pre-trained on AudioSet. **SSL+SL** denotes the SSL pre-trained BEATs model is additionally supervised fine-tuned on AudioSet. * indicates the encoder of BEATs is frozen during the fine-tuning on ASC dataset. Top-1 accuracy of 5 independent runs is presented.

Before fine-tuning for ASC, the reserved BEATs encoder is appended with a task-specific linear classifier to output class probabilities for different acoustic scenes, as shown in Figure 1 (b). The linear classifier consists of a linear layer, a mean-pooling layer and a softmax operation. The fine-tuning data is from TAU Urban Acoustic Scene 2022 Mobile development dataset [14]. Each audio clip is resampled to 16 kHz, and 128-dimensional Mel-filter bank features are extracted using a 25 ms Povey window with a 10 ms shift. The features are normalized according to the mean and standard deviation of AudioSet. Each acoustic feature $x \in \mathbb{R}^{F\times T}$ is then divided into 16 × 16 patches and flattened into a sequence of patches to serve as input for the pre-trained BEATs model.

2.2. Frozen Fine-tuning

To evaluate the benefits of self-supervised audio representations, the encoder of BEATs (SSL) is frozen as a feature extractor while only the linear classifier is trained with the cross entropy loss, as shown in Figure 1 (b). The frozen model is denoted as BEATs (SSL)*. Frozen fine-tuning allows the model to leverage representations learned during self-supervised pre-training, preventing overfitting and catastrophic forgetting [15]. We train BEATs (SSL)* for 60 epochs using the default Adam optimizer. To further enhance the robustness and generalization of the model, we apply two widely-used data augmentation methods: Mixup [16] with an α of 0.3 and SpecAugmentation [17] with a mask ratio of 0.2.

2.3. Unfrozen Fine-tuning

Beside freezing the SSL models as feature extractors, we also explore unfrozen fine-tuning to further adapt BEATs to the ASC task. Unfrozen fine-tuning allows the model to refine representations learned during self-supervised pre-training, typically leading to better performance compared to frozen fine-tuning.

We apply BEATs (SSL) and BEATs (SSL+SL) for unfrozen fine-tuning, using the same training configurations. The models are fine-tuned for 30 epochs with a batch size of 512. The AdamW optimizer [18] is applied with β = (0.9, 0.98) and a weight decay of 0.01. The learning rate is scheduled to exponentially increase from 0 to a peak value of 1×10^{-5} over four epochs, then linearly decrease to a minimum value of 5×10^{-8} for the remaining epochs. Four data augmentation techniques are used during fine-tuning: Mixup [16] with α = 0.3, Freq-MixStyle [19] with α = 0.4 and p_{fms} = 0.4, SpecAugmentation [17] with a mask ratio of 0.2, and DIR augmentation [20] with p_{dir} = 0.6.

2.4. Ensemble Models

Previous works [3, 19] have shown that model ensemble with different configurations can enhance ASC performance and benefit knowledge distillation. In this work, we average the logits to ensemble BEATs models that fine-tuned with different fine-tuning strategies. The small ensemble consists of three fine-tuned BEATs models: BEATs (SSL)*, BEATs (SSL) and BEATs (SSL+SL). The large ensemble includes twelve fine-tuned BEATs models: one BEATs (SSL)*, one BEATs (SSL) and ten BEATs (SSL+SL). The ten BEATs (SSL+SL) models are AudioSet fine-tuned BEATs models with different tokenizers as described in the original work [10].

3. KNOWLEDGE DISTILLATION WITH SELF-SUPERVISED TEACHERS

DCASE Challenge 2024 task 1 imposes strict limitations on computational complexity, restraining the model size within 128kB and the number of multiply-accumulate operations within 30 MMACs. In this section, knowledge distillation [21] is introduced to transfer knowledge from the fine-tuned BEATs to an efficient student model, TF-SepNet-64. By employing the self-supervised teachers, we aim to develop ASC systems that operate within the computational limits while maintaining high accuracy with limited labeled data. The framework of proposed system is shown in Figure 1 (c).

3.1. TF-SepNet-64

Time-Frequency Separate Network (TF-SepNet) [12] is a deep CNN architecture designed specifically for low-complexity ASC tasks, achieving second place in DCASE Challenge 2023. TF-SepNet processes features separately along the time and frequency dimensions using one-dimensional (1D) kernels, which reduce computational costs and provide a larger effective receptive field (ERF), allowing the model to capture more time-frequency features.

As in [13], TF-SepNet-64 is optimized to meet the upper complexity limit of the challenge requirements. Several adjustments have been made. First, the number of base channels is set to 64. Second, all Adaptive Residual Normalization layers [4] are replaced with Residual Normalization layers [2] to reduce the number of model parameters. Third, a Max-pooling layer is added before the last TF-SepConvs block to further reduce the feature size. In the finish, the total parameter number of TF-SepNet-64 is 126,858. For an input feature size of (512, 64), the maximum number of MACs per inference is 29.4196 MMACs.

3.2. Knowledge Distillation

We adopt the widely used knowledge distillation framework in previous years' challenges [3, 19], which focuses on directly mimicking the final predictions of the teacher model. As illustrated in Figure 1 (c), the knowledge transfer involves two main steps.

The input feature is a log-mel spectrogram $x \in \mathbb{R}^{F\times T}$. For the teacher path, once the self-supervised teachers are fine-tuned, as shown in Figure 1 (b), the predictions on a specified training subset are computed, serving as the teacher logits in the knowledge distillation process. For the student path, the ASC student is trained on the specified training subset using a combination of the ground truth labels and the soft targets provided by the teacher model. Give a vector of logits z as the outputs of the last classification layer of a model, the soft targets are the probabilities that the input belongs to the classes and can be estimated by a softmax function $\delta(\cdot)$ as

$$\delta(z_i, \tau) = \frac{\exp(z_i/\tau)}{\sum_j \exp(z_j/\tau)} \quad (1)$$

where z_i is the logit for the i-th class, and a temperature factor τ is introduced to control the importance of each soft target. The training objective of student model is to minimize the divergence between the student's predictions and the soft targets from the teacher, as well as to correctly classify the labeled data. The overall loss function for the student can be formulated as

$$\mathcal{L} = \lambda\mathcal{L}_{CE}(y, \delta(z_s)) + (1-\lambda)\tau^2\mathcal{L}_{KL}(\delta(z_t, \tau), \delta(z_s, \tau)) \quad (2)$$

where $\mathcal{L}_{CE}$ is the cross-entropy loss between the ground truth labels and the student's predictions, and $\mathcal{L}_{KL}$ is the Kullback-Leibler divergence between the soft targets from the teacher and the student's predictions. λ is a hyperparameter to balance the weight between label and distillation loss.

3.3. Experimental Setup

Dataset and Baseline: The dataset for the task1 of DCASE 2024 Challenge has exactly the same content as the TAU Urban Acoustic Scenes 2022 Mobile development dataset [14], but the training sets of different sizes are provided. These train subsets contain approximately 5%, 10%, 25%, 50%, and 100% of the audio snippets in the training set provided in previous years. The DCASE baseline model for comparison, CP-Mobile [22], is a fully-supervised CNN classifier that achieved top ranking in DCASE Challenge 2023.
Feature Extraction: For TF-SepNet-64, we generally follow the baseline settings [22] for feature extraction. The audio recordings are firstly resampled to 32 kHz. Time-frequency representations are then extracted using a 4096-point FFT with a window size of 96 ms and a hop size of 16 ms. The primary difference in our approach is the application of a Mel-scaled filter bank with a large number of frequency bins, 512, to convert the spectrograms into mel spectrograms, which leads to a slight improvement on the classification accuracy. The final input size for TF-SepNet-64 is (512, 64).
Data Augmentations: Data augmentation is a crucial technique in ASC tasks, especially when the labeled data is limited. In this work, we use a combination of Soft Mixup [13], Freq-MixStyle [19], and Device Impulse Response (DIR) augmentation [20] to enhance the diversity and quality of our training data. α of Soft Mixup is set to 0.3. α and p of Freq-MixStyle are respectively set to 0.4 and 0.8. p_{dir} of DIR augmentation is set to 0.4. All augmentations are implemented to be plug-and-played during training.
Training: We train TF-SepNet-64 for 150 epoch using Adam optimizer with different initial learning rate for 5 subsets, 0.06 for split5, 0.05 for split50 and 0.04 for all other splits. Stochastic Gradient Descent with Warm Restarts (SGDR) [23] is applied with T_0 =10 and T_{mult} = 2, where the learning rate is periodically reset to initial value and then decayed with cosine annealing. The batch size is set to 512. We fix $\lambda = 0.02$ and $\tau = 2$ for the knowledge distillation as in [3]. After training, Post-Training Static Quantization is implemented through the Intel Neural Compressor[2] to quantize the weights of model into INT8 data type.

[2] https://intel.github.io/neural-compressor

Model	5%	10%	25%	50%	100%	Avg.
DCASE Baseline	42.4	45.3	50.3	53.2	57.0	49.6
TF-SepNet-64	45.7	51.1	55.6	59.6	62.5	54.9
+BEATs (SSL)*	48.2	51.0	54.9	58.0	59.9	54.4
+BEATs (SSL)	47.3	**52.5**	57.6	60.8	61.9	56.0
+BEATs (SSL+SL)	47.8	52.1	57.7	**61.1**	62.6	56.3
+3 Ensemble	**49.0**	52.3	**57.9**	60.7	**63.5**	**56.7**
+12 Ensemble	47.9	52.3	57.5	60.1	62.8	56.1

Table 2: Accuracy of TF-SepNet-64 with different BEATs teachers on the test set of TAU Urban Acoustic Scene 2022 Mobile development dataset [14]. The teacher logits of each BEATs model is used (**+**) in knowledge distillation at a time. Top-1 and quantized accuracy of 5 independent runs is presented.

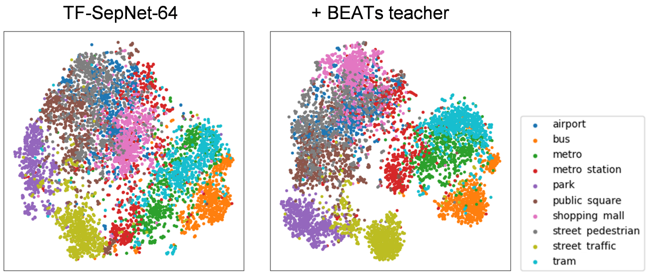


Figure 2: TSNE [24] visualization of acoustic scene features extracted by TF-SepNet-64, which is trained on the 5% subset. **Left:** Knowledge distillation is not applied. **Right:** Distilling knowledge from the 3 ensemble BEATs teacher.

4. RESULTS

4.1. Performance of Fine-tuned BEATs

Table 1 presents the accuracy of fine-tuned BEATs using different fine-tuning strategies. Even with the encoder frozen, BEATs (SSL)* achieves over 50% accuracy with only 5% training data. This result demonstrates the self-supervised representations learned from general-purpose audio dataset are beneficial to the ASC task, especially when labeled data is exceptionally limited. However, the accuracy witnesses little improvements with the increase of training data. This is due to the limited capability of a single linear layer to adapt to changes in data scale. When the encoder is unfrozen during fine-tuning, BEATs (SSL) shows a significant 3.9% improvement in average accuracy. Additionally, the AudioSet supervised fine-tuned model, BEATs (SSL+SL), achieves further improvements. For the model ensembles, the 3 ensemble outperforms the best single model by 1.4% in average accuracy, and the large 12 ensemble achieves an average accuracy of 60.6%. The different fine-tuning strategies diversify the predictions for ensembling, effectively combining self-supervised knowledge and supervised knowledge.

4.2. TF-SepNet-64 with BEATs Teachers

The performance of TF-SepNet-64 with various BEATs teachers is shown in Table 2. TF-SepNet-64 without knowledge distillation outperforms the DCASE baseline by 5.3% in average accuracy but experiences considerable drop as the amount of training data decreases. The single BEATs (SSL)* teacher only helps in the 5% subset while BEATs (SSL) and BEATs (SSL+SL) improve the student model across more subsets. By comparing the performance between TF-SepNet-64 and BEATs, we infer that a teacher model is generally helpful when it has a higher accuracy than the student. Nevertheless, BEATs (SSL) helps to obtain the highest accuracy in the 10% subset while BEATs (SSL+SL) is most effective in the 50% subset. Compared to individual teachers, the ensemble teachers generally provide greater benefits to the student. Interestingly, rather than the large 12 ensemble, the small 3 ensemble achieves the best performance for the remaining subsets, obtaining the highest average accuracy of 56.7%. Therefore, a teacher with higher accuracy does not necessarily guarantee better improvement for the student. To further examine the benefits of BEATs teacher, we visualize the acoustic scene features as shown in Figure 2. The samples are better clustered with the assistance of BEATs teacher.

System	5%	100%	MMACs	Param/k
Proposed System	**49.0**	**63.5**	29.4	126.9
Mel bins (512→256)	47.3	61.9	14.8	126.9
Base channels (64→40)	45.8	61.3	12.9	52.3
ResNorm→AdaResNorm	48.8	61.9	29.4	128.6
w/o added Max-pooling	45.9	63.3	32.0	126.9
w/o Soft Mixup	46.0	62.3	29.4	126.9
w/o Freq-MixStyle	47.0	61.6	29.4	126.9
w/o DIR Augmentaion	48.0	62.4	29.4	126.9
w/o BEATs teacher	45.7	62.5	29.4	126.9

Table 3: Ablation study of our proposed system (TF-SepNet-64 + 3 BEATs ensemble). Each component is changed (→) or removed (**w/o**) at a time. **MMACs** (million multiply-accumulate operations) represents the computational costs per inference. **Param/k** denotes the number of parameters.

4.3. Ablation Study

Table 3 presents the ablation study for our proposed system (TF-SepNet-64 + 3 BEATs ensemble) on the two extreme subset: 5% and 100%. The configurations for TF-SepNet-64, such as using a larger amounts of Mel bins, more base channels, replacing AdaResNorm with ResNorm, and adding a Max-pooling layer, contributes to performance improvements to varying degrees while maintaining the system's complexity within the challenge requirements. Meanwhile, the data augmentation methods enhance the accuracy without introducing additional overheads. The results also indicate that the BEATs teacher is the dominant factor in performance when labeled training data is extremely limited.

5. CONCLUSION

In this paper, we introduce self-supervised audio representations to address the challenge of data-efficient low-complexity acoustic scene classification (ASC). We fine-tune BEATs models as self-supervised teachers and then transfer the knowledge to a low-complexity student model, TF-SepNet-64, through a knowledge distillation framework. The experimental results demonstrate the effectiveness of self-supervised pre-trained models in the ASC task, and also show the benefits of self-supervised teachers for the low-complexity student model when the labeled training data is limited.

6. ACKNOWLEDGEMENT

This project is supported by the Gusu Innovation and Entrepreneurship Leading Talents Programme (No: ZXL2022472).

7. REFERENCES

[1] D. Barchiesi, D. Giannoulis, D. Stowell, and M. D. Plumbley, "Acoustic scene classification: Classifying environments from the sounds they produce," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 16–34, 2015.

[2] B. Kim, S. Yang, J. Kim, and S. Chang, "QTI submission to DCASE 2021: Residual normalization for device-imbalanced acoustic scene classification with efficient design," DCASE2021 Challenge, Tech. Rep., June 2021.

[3] F. Schmid, T. Morocutti, S. Masoudian, K. Koutini, and G. Widmer, "CP-JKU submission to DCASE23: Efficient acoustic scene classification with cp-mobile," DCASE2023 Challenge, Tech. Rep., May 2023.

[4] Y. Cai, M. Lin, C. Zhu, S. Li, and X. Shao, "DCASE2023 task1 submission: Device simulation and time-frequency separable convolution for acoustic scene classification," DCASE2023 Challenge, Tech. Rep., May 2023.

[5] Y. Tan, H. Ai, S. Li, and M. D. Plumbley, "Acoustic scene classification across cities and devices via feature disentanglement," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.

[6] F. Schmid, P. Primus, T. Heittola, A. Mesaros, I. Martín-Morató, K. Koutini, and G. Widmer, "Data-efficient low-complexity acoustic scene classification in the dcase 2024 challenge," *arXiv preprint arXiv:2405.10018*, 2024.

[7] Y. Gong, C.-I. Lai, Y.-A. Chung, and J. Glass, "Ssast: Self-supervised audio spectrogram transformer," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 10, 2022, pp. 10 699–10 709.

[8] P.-Y. Huang, H. Xu, J. Li, A. Baevski, M. Auli, W. Galuba, F. Metze, and C. Feichtenhofer, "Masked autoencoders that listen," *Advances in Neural Information Processing Systems*, vol. 35, pp. 28 708–28 720, 2022.

[9] D. Chong, H. Wang, P. Zhou, and Q. Zeng, "Masked spectrogram prediction for self-supervised audio pre-training," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[10] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, Z. Chen, W. Che, X. Yu, and F. Wei, "BEATs: Audio pre-training with acoustic tokenizers," in *Proceedings of the 40th International Conference on Machine Learning (ICML)*. PMLR, 23–29 Jul 2023, pp. 5178–5193.

[11] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 776–780.

[12] Y. Cai, P. Zhang, and S. Li, "TF-SepNet: An efficient 1D kernel design in CNNs for low-complexity acoustic scene classification," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 821–825.

[13] Y. Cai, M. Lin, S. Li, and X. Shao, "DCASE2024 task1 submission: Data-efficient acoustic scene classification with self-supervised teachers," DCASE2024 Challenge, Tech. Rep., May 2024.

[14] T. Heittola, A. Mesaros, and T. Virtanen, "Acoustic scene classification in DCASE 2020 challenge: generalization across devices and low complexity solutions," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020)*, 2020, pp. 56–60.

[15] M. Davari, N. Asadi, S. Mudur, R. Aljundi, and E. Belilovsky, "Probing representation forgetting in supervised and unsupervised continual learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 16 712–16 721.

[16] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[17] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," in *Proceedings of the Conference of the International Speech Communication Association (INTERSPEECH)*. ISCA, 2019.

[18] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations (ICLR)*, 2019.

[19] F. Schmid, S. Masoudian, K. Koutini, and G. Widmer, "CP-JKU submission to DCASE22: Distilling knowledge for low-complexity convolutional neural networks from a patchout audio transformer," DCASE2022 Challenge, Tech. Rep., June 2022.

[20] T. Morocutti, F. Schmid, K. Koutini, and G. Widmer, "Device-robust acoustic scene classification via impulse response augmentation," in *2023 31st European Signal Processing Conference (EUSIPCO)*. IEEE, 2023, pp. 176–180.

[21] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *International Journal of Computer Vision*, vol. 129, no. 6, pp. 1789–1819, 2021.

[22] F. Schmid, T. Morocutti, S. Masoudian, K. Koutini, and G. Widmer, "Distilling the knowledge of transformers and CNNs with CP-mobile," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2023 Workshop (DCASE2023)*, 2023, pp. 161–165.

[23] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[24] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

LANGUAGE-QUERIED AUDIO SOURCE SEPARATION ENHANCED BY EXPANDED LANGUAGE-AUDIO CONTRASTIVE LOSS

Hae Chun Chung

KT Corporation
Acoustic Processing Project, AI Tech Lab
Seoul, Republic of Korea
hc.chung@kt.com

Jae Hoon Jung

KT Corporation
Acoustic Processing Project, AI Tech Lab
Seoul, Republic of Korea
hoony.jung@kt.com

ABSTRACT

Audio sources recorded for specific purposes often contain extraneous sounds that deviate from the intended goal. Re-recording to achieve the desired result is expensive. However, separating the target source from the original audio source based on natural language queries would be much more efficient. However, audio source separation with natural language queries is a complex task. To address this, the DCASE 2024 Challenge Task 9 proposed language-queried audio source separation (LASS). This paper aims to tackle LASS by proposing an extended language-audio contrastive learning approach. To align the separated output audio with the target text and target audio, we first designed audio-to-text contrastive loss and audio-to-audio contrastive loss, respectively. By leveraging the characteristics of contrastive learning, we combined these two losses into an extended audio-to-multi contrastive loss. Our model, trained with this loss, improves the signal-to-distortion ratio (SDR) by more than 30% compared to the baseline provided by the challenge.

Index Terms— Source Separation, Contrastive Learning

1. INTRODUCTION

In real-world scenarios, unintended and uncontrollable events frequently occur. During on-location content recording, numerous factors are managed to capture the desired purposes. Nevertheless, unwanted elements often contaminate the recorded audio sources. Re-recording to achieve perfection is not only expensive but also challenging. If an AI model could separate the target source from the recorded audio source based on natural language queries, these costs could be significantly reduced. However, audio source separation with natural language queries is a complex task. Consequently, research in this field is limited, and existing performance levels are suboptimal [1, 2]. To address this, the DCASE 2024 Challenge Task 9 proposed language-queried audio source separation (LASS) [3]. This task focuses on developing a system that separates the target audio source from a mixed audio source based on a text description about the intended audio.

LASS-Net [1] first introduced the task of language-queried audio source separation (LASS), proposing an end-to-end neural network consisting of a text encoder, which takes a text description (target text) as input and outputs a text embedding, and a separator, which takes the mixed audio (a mixture of a target audio and a noise audio) and text embedding as inputs to predict the target audio. AudioSep [2] used a contrastive language-audio pre-training model (CLAP) model [4] as the text encoder, which was frozen during training, and the separator was trained to predict phase residuals as well as a magnitude mask [5]. Furthermore, unlike LASS-Net, which was trained on a subset of the AudioCaps dataset [6], AudioSep was trained with large-scale audio datasets, leading to a significant performance improvement over LASS-Net. The baseline system for the DCASE 2024 Challenge Task 9 is based on the AudioSep model, but it only used the development set (Clotho [7] and augmented FSD50K [8] dataset) provided in the challenge for training data. This baseline system achieved a signal-to-distortion ratio (SDR) score of 5.708 when evaluated on the validation dataset provided in the challenge.

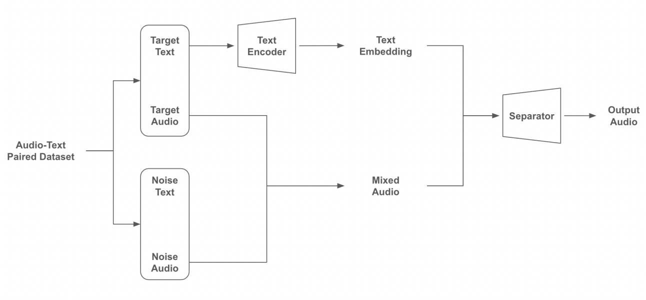


Figure 1: The figure above provides a schematic overview of our model. From the audio-text paired dataset, a pair consisting of target text and target audio, along with a pair of noise text and noise audio, are randomly sampled to ensure they do not overlap. The target audio and noise audio are mixed at a signal-to-noise ratio (SNR) ranging from -15 to 15 dB to create a mixed audio. The text encoder extracts a text embedding from the target text. The separator then takes this text embedding as a condition and the mixed audio, separating the output audio conditioned on the text embedding from the mixed audio.

We also adopted a model structure consisting of a text encoder and a separator. The separator was same to ResUNet [5] setting used in AudioSep. For the text encoder, we used FLAN-T5 [9], an instruction-tuned large language model (LLM), instead of the CLAP model. FLAN-T5 was chosen based on its successful application as a text encoder in TANGO [10], which addresses the text-to-audio generation task. To train this system, we introduced three loss functions, and utilized a loss balancer [11] to stabilize the training. First, L1 loss was employed to align the separated audio waveform with the target audio waveform in the time domain. Second, to optimize performance in both the time and frequency domains, we utilized multi-scale mel-spectrogram loss [12, 13, 11, 14], applied across multiple time scales in the mel-spectrogram. Lastly, contrastive loss was introduced in addition to L1 loss and spectrogram loss.We designed three distinct contrastive losses using target audio, noise audio, target text, and noise text for output audio. To embed audio and text, CLAP model [4] was used. First, audio-to-text contrastive loss (A2T-CL) was introduced to increase the similarity between output audio and target text while reducing the similarity with other non-target texts within the mini-batch. The performance was further improved by combining audio-to-audio contrastive loss (A2A-CL), which applies to the target audio and other non-target audios within the mini-batch, with A2T-CL. Contrastive learning tends to improve performance as the comparison samples, especially negative samples, increases [15, 16]. For leveraging this, we designed the audio-to-multi contrastive loss (A2M-CL) by integrating A2A-CL and A2T-CL into a single expanded loss. A2M-CL encourages output audio to increase similarity for both the target text and the target audio while reducing similarity for other non-target texts and audios in the mini-batch. This doubles the number of comparison samples, both positive and negative samples, than A2A-CL or A2T-CL. We experimented for each method and achieved SDR scores of 7.030, 7.12, and 7.139, respectively. This is a performance improvement of more than 30% over the baseline model.

2. METHODS

2.1. Overview

Our system consists of two models: a text encoder and a separator. For the text encoder, we utilize FLAN-T5 [9], an enhanced version of the text-to-text transfer transformer (T5) model [17]. FLAN-T5 is initialized with a T5 checkpoint and fine-tuned with instructions and chain-of-thought reasoning, enabling it to extract robust text embeddings from text descriptions with its strong text representation capacity. TANGO [10], which tackles the text-to-audio generation task, demonstrated effectiveness of FLAN-T5 as the text encoder for cross-modal task.

The separator is the ResUNet model [18, 5], an advanced version of the UNet model. We used the same setting as ResUNet used in AudioSep [2]. The ResUNet model takes a mixed audio waveform and text embedding as input and separates the output audio waveform conditioned on the text from the mixed audio. The process begins with applying a short-time Fourier transform (STFT) to the waveform to extract the complex spectrogram, magnitude spectrogram and phase. The ResUNet model takes the complex spectrogram and outputs the magnitude mask and phase residual conditioned on the text embedding. The separated complex spectrogram is obtained by multiplying the STFT of the mixture with the predicted magnitude mask and phase residual. Finally, the separated complex spectrogram is converted back into an audio waveform using the inverse short-time Fourier transform (iSTFT).

2.2. Training Loss Terms

From the audio-text paired dataset, N target pairs (target audio d^{ta} and target text d^{tt}) and N noise pairs (noise audio d^{nt} and noise text d^{nt}) are randomly sampled to ensure they do not overlap. For creating mixed audio waveform d^{ma}, two audio waveforms are combined with a signal-to-noise ratio (SNR) ranging from -15 to 15 dB. The target text is forwarded into the text encoder to extract the text embedding. The separator then takes the mixed audio waveform and the text embedding, separating the output audio waveform d^{oa} conditioned on the text from the mixture.

L1 Loss In the source separation task, it is crucial to extract the desired target sound source from a given mixture without altering its original characteristics. In other words, the closer the separated sound source is to the target sound source, the better the performance. To achieve this, minimizing the L1 distance between the target audio and separated audio over the time domain is commonly used due to its simplicity and effectiveness in universal source separation tasks. We also applied this approach. The equation is as follows:

$$\mathcal{L}_{time} = \left\|d^{ta} - d^{oa}\right\|_1 \tag{1}$$

Spectrogram Loss To optimize performance in both the time and frequency domains, we also employed a multi-scale mel-spectrogram loss [12, 13, 11, 14] applied across multi time scales in the mel-spectrogram. This loss is calculated based on the distance in the mel-spectrogram, which is derived from the short-time Fourier transform (STFT) and converted to a mel scale that better captures human auditory characteristics. This approach enhances the perceptual quality of the output. Additionally, using loss functions on mel-spectrograms across multiple STFT scales enables the model to effectively capture the time-frequency distribution, significantly enhancing its overall performance.

$$\begin{aligned}\mathcal{L}_{freq} = \frac{1}{|\alpha| + |s|} \sum_{\alpha_i \in \alpha} \sum_{i \in e} \left\|\mathcal{S}_i(d^{ta}) - \mathcal{S}_i(d^{oa})\right\|_1 \\ + \alpha_i \left\|\log \mathcal{S}_i(d^{ta}) - \log \mathcal{S}_i(d^{oa})\right\|_2\end{aligned} \tag{2}$$

where $\mathcal{S}_i$ is a 64-bins mel-spectrogram using a normalized STFT with window size of 2^i and hop length of 2^{i-1}, $e = 6, ..., 12$ is the set of scales, and α represents the set of scalar coefficients balancing between the L1 and L2 terms, $\alpha_i = \sqrt{2^{i-1}}$. Here, $|\alpha|$ denotes the sum of the elements of the α set, and $|s|$ is the number of scales.

Audio-to-Text Contrastive Loss The output audio of the text-conditioned audio source separation should match the target audio, for which L1 loss and spectrogram loss were used. Additionally, the output audio must involve all the content of the target text while excluding any content not present in the target text. To achieve this, we implemented an audio-to-text contrastive loss (A2T-CL) using the contrastive language-audio pre-training (CLAP) model [4]. CLAP was trained to align audio and text by projecting them into a shared feature space. Firstly, we designed the loss so that the output audio attracts its corresponding target text as positive and repels other target texts within the mini-batch as negative in the shared feature space of CLAP model. Contrastive learning tends to improve performance as the comparison samples, especially negative samples, increases [15, 16]. To leverage this, we additionally use noisy texts within the mini-batch as negative examples. This approach encourages the output audio to be distinguishable from various other texts while accurately fitting the target text. The equation is as follows:

$$\mathcal{L}_{a2t} = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{\exp(f_i^{oa} \cdot f_i^{tt}/\tau)}{\sum\limits_{k=1}^{N} \left\{\exp(f_i^{oa} \cdot f_k^{tt}/\tau) + \exp(f_i^{oa} \cdot f_k^{nt}/\tau)\right\}} \tag{3}$$

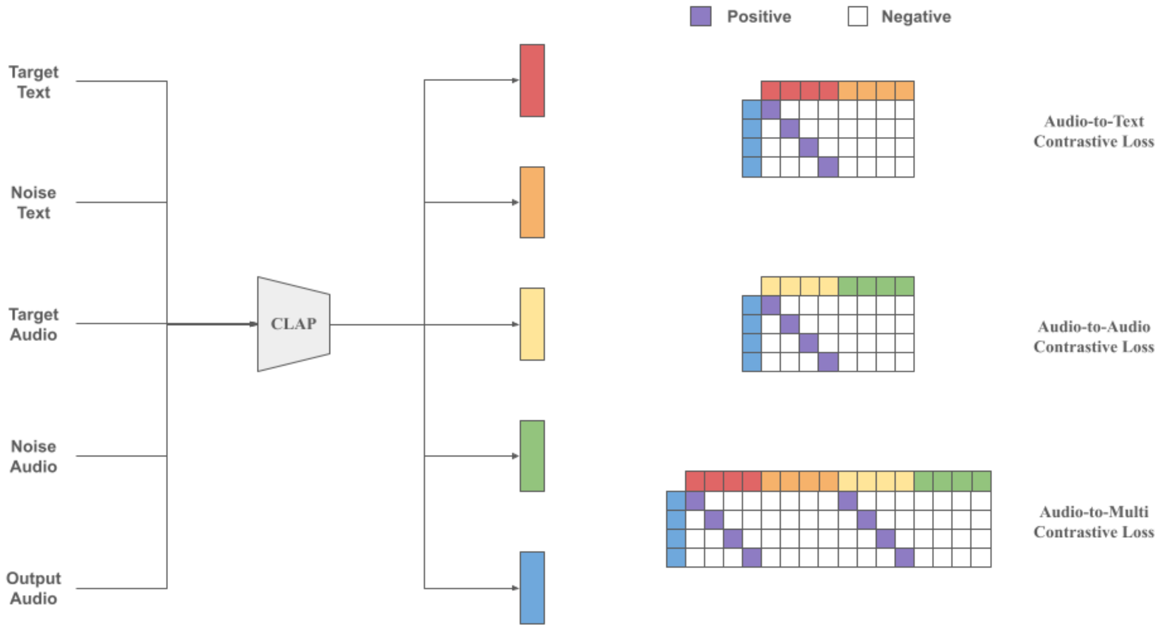

Positive
Negative
Target Text
Noise Text
Target Audio
Noise Audio
Output Audio
CLAP
Audio-to-Text Contrastive Loss
Audio-to-Audio Contrastive Loss
Audio-to-Multi Contrastive Loss

Figure 2: Each rectangle in red, orange, yellow, green, and blue represents the features of target texts, noise texts, target audios, noise audios, and the output audios of the separator, all embedded using the CLAP model. The matrices on the right schematically illustrate three types of contrastive loss with a mini-batch size of 4. In these matrices, purple spaces indicate positive relationships, while white spaces indicate negative relationships. The output audio has positive relationships with its corresponding target text and target audio, whereas all other texts and audios within the mini-batch are considered negative relationships.

where f^{oa} is a feature with output audio embedded using audio encoder of CLAP model, and f^{tt} and f^{nt} are features with target text and noise text embedded using text encoder of CLAP model. And τ is a scalar temperature parameter.

Audio-to-Audio Contrastive Loss The concept of A2T-CL, which encourages output audio to contain only the content of the target text, can also be applied to audios (target audios and noise audios). Therefore, it is possible to design an audio-to-audio contrastive loss (A2T-CL) using these.

$$\mathcal{L}_{a2a} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp(f_i^{oa} \cdot f_i^{ta}/\tau)}{\sum\limits_{k=1}^{N} \left\{\exp(f_i^{oa} \cdot f_k^{ta}/\tau) + \exp(f_i^{oa} \cdot f_k^{na}/\tau)\right\}} \tag{4}$$

where f^{ta} and f^{na} are features with target audio and noise audio embedded using audio encoder of CLAP model.

Audio-to-Multi Contrastive Loss As aforementioned, contrastive learning shows better performance as the number of the comparison samples increases. To take advantage of this, we integrated audio-to-text contrastive loss and audio-to-audio contrastive loss into a single expanded loss: audio-to-multi contrastive loss (A2M-CL), effectively doubling the number of the comparison samples. This causes the output audio to pull closer to its corresponding target text and target audio while pushing away from all remaining target texts, noise texts, target audios, and noise audios within the mini-batch. As a result, the output audio maximizes its similarity to both the target text and target audio.

$$a2t_i = \sum_{k=1}^{N} \left\{\exp(f_i^{oa} \cdot f_k^{tt}/\tau) + \exp(f_i^{oa} \cdot f_k^{nt}/\tau)\right\} \tag{5}$$

$$a2a_i = \sum_{k=1}^{N} \left\{\exp(f_i^{oa} \cdot f_k^{ta}/\tau) + \exp(f_i^{oa} \cdot f_k^{na}/\tau)\right\} \tag{6}$$

$$\mathcal{L}_{a2m} = -\frac{1}{N}\sum_{i=1}^{N} \frac{1}{2}\left\{\log \frac{\exp(f_i^{oa} \cdot f_i^{tt}/\tau)}{a2t_i + a2a_i} + \log \frac{\exp(f_i^{oa} \cdot f_i^{ta}/\tau)}{a2t_i + a2a_i}\right\} \tag{7}$$

Loss Balancer Encodec [11] introduced a loss balancer to stabilize the training by adjusting the loss weights based on various scales of gradients from the model. We used a loss balancer to stabilize the model training with various losses. The gradient $\frac{\partial l_i}{\partial d^{oa}}$ of the loss based on the output d^{oa} is recalculated using the following equation, incorporating the weights λ_i for the loss and reference norm R.

$$\tilde{g}_i = R \frac{\lambda_i}{\sum_j \lambda_j} \cdot \frac{g_i}{\left\langle \|g_i\|_2 \right\rangle_\beta} \tag{8}$$

where $\left\langle \|g_i\|_2 \right\rangle_\beta$ is the exponential moving average of g_i. We take $R = 1$ and $\beta = 0.999$. All the model losses fit into the balancer. The model is then backpropagated to $\sum_i \tilde{g}_i$ instead of the original $\sum_i \lambda_i g_i$.

2.3. Proposed Systems

We propose a total of three systems. The process by which data is preprocessed and fed forward to the model in all systems is the same as mentioned in Section 2.1. The primary difference between each system lies in the configuration of losses during the training process, particularly the type of contrastive loss. The configuration of the losses for each system in our training was defined as follows. All weights λ for the losses are set 1.

$$System_1 = \lambda_1 \mathcal{L}_{time} + \lambda_2 \mathcal{L}_{freq} + \lambda_3 \mathcal{L}_{a2t} \tag{9}$$

$$System_2 = \lambda_1 \mathcal{L}_{time} + \lambda_2 \mathcal{L}_{freq} + \lambda_3 \mathcal{L}_{a2t} + \lambda_4 \mathcal{L}_{a2a} \tag{10}$$

$$System_3 = \lambda_1 \mathcal{L}_{time} + \lambda_2 \mathcal{L}_{freq} + \lambda_3 \mathcal{L}_{a2m} \tag{11}$$

3. SETTING

3.1. Training Data

A total of four datasets were used for model training: AudioCaps [6], WavCaps [19], Clotho v2 [7], and FSD50K [8]. For the WavCaps dataset, only data belonging to AudioSet were used. The combined dataset comprises a total of 216,398 audio clips, amounting to approximately 580 hours. The following procedure was employed to generate mixed audio:

1. Random Selection: Target and noise audio clips were randomly selected to ensure no overlap within the entire dataset.
2. Mono Conversion: If an audio clip had 2 channels, the average of the two channels was calculated to convert it into a mono clip.
3. Resampling: Audio clips with a sampling rate different from 16 kHz were resampled to 16 kHz.
4. Length Adjustment: If an audio clip exceeded 10 seconds in length, it was randomly truncated to 10 seconds. If it was shorter than 10 seconds, zero padding was added to the end to make it 10 seconds long.
5. Mixing: The pre-processed target audio clip and a noise audio clip were mixed with signal-to-noise ratios (SNR) ranging from -15 dB to 15 dB to produce a mixed audio clip.

3.2. Model

The text encoder for embedding the text is used pre-trained FLAN-T5 model [9], and all parameters were frozen. AdamW optimizer [20] with a learning rate of 0.0003 is used for training the separator with the batch size of 25. τ was all set to 0.1 for the contrastive loss.

3.3. Test Data

To evaluate the performance of the model, validation (synth) dataset provided in DCASE2024 Challenge Task9 [3] was used.

3.4. Metric

To compare the performance of language-queried audio source separation (LASS), we used three objective metrics that are commonly used in the field of source separation: signal-to-distortion ratio (SDR), signal-to-distortion ratio enhancement (SDRi), and scale-invariant SDR (SI-SDR) [21].

4. RESULTS

The language-queried audio source separation (LASS) task is a nascent field with limited prior research. However, due to its high usability and future potential, the DCASE Challenge adopted this task as Task 9 for this year. We participated in Task 9 of the DCASE 2024 Challenge to officially demonstrate the performance of our model, specifically designed for LASS

	SDR	SDRi	SI-SDR
Baseline	5.708	5.673	3.862
System1	7.030	6.995	5.368
System2	7.124	7.089	5.593
System3	7.139	7.104	5.504

Table 1: The comparison of baseline model and our proposed model on validation set.

We compared our system with the baseline model provided by challenge. The baseline provided in the challenge was based on the AudioSep model. Table 1 shows the performance comparison between the baseline model and our proposed systems using the challenge validation set. Our proposed systems show significant performance improvements across all three metrics. While the baseline provided for the challenge achieved a signal-to-distortion ratio (SDR) score of 5.708, our systems achieved SDR scores of 7.030, 7.124, and 7.139, respectively. This represents a remarkable performance improvement of over 30% compared to the baseline. In language-queried audio source separation (LASS), it is crucial to precisely match the output audio to the target audio. Additionally, we demonstrate that aligning the output audio more closely with both the target text and target audio in the feature space using contrastive learning enhances performance. We also show the effectiveness of the audio-to-multi contrastive loss, which leverages the characteristics of contrastive learning by integrating audio-to-text and audio-to-audio contrastive losses. This approach leverages the advantage of having more negatives, significantly improving the model's effectiveness.

	SDR	SDRi	SI-SDR
Baseline	5.799	5.693	3.873
System1	7.302	7.195	5.628
System2	7.186	7.080	5.526
System3	7.118	7.012	5.301

Table 2: The comparison of baseline model and our proposed model on evaluation set.

We received a score by submitting the results of each system on the evaluation set to the challenge. Contrary to expectations, the evaluation in the evaluation set came out opposite to the evaluation in the validation set. We doubt whether this is an overfitting on the validation set. We leave it as a future work. However, nevertheless, the systems we proposed showed significant performance improvement compared to the baseline. The performance can be improved using the contrastive learning method we designed simply without modification to the model architecture.

5. REFERENCES

[1] X. Liu, H. Liu, Q. Kong, X. Mei, J. Zhao, Q. Huang, M. D. Plumbley, and W. Wang, "Separate what you describe: Language-queried audio source separation," *arXiv preprint arXiv:2203.15147*, 2022.

[2] X. Liu, Q. Kong, Y. Zhao, H. Liu, Y. Yuan, Y. Liu, R. Xia, Y. Wang, M. D. Plumbley, and W. Wang, "Separate anything you describe," *arXiv preprint arXiv:2308.05037*, 2023.

[3] https://dcase.community/challenge2024/task-language-queried-audio-source-separation.

[4] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, "Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[5] Q. Kong, K. Chen, H. Liu, X. Du, T. Berg-Kirkpatrick, S. Dubnov, and M. D. Plumbley, "Universal source separation with weakly labelled data," *arXiv preprint arXiv:2305.07447*, 2023.

[6] C. D. Kim, B. Kim, H. Lee, and G. Kim, "Audiocaps: Generating captions for audios in the wild," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 119–132.

[7] K. Drossos, S. Lipping, and T. Virtanen, "Clotho: An audio captioning dataset," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 736–740.

[8] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "Fsd50k: an open dataset of human-labeled sound events," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 829–852, 2021.

[9] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, *et al.*, "Scaling instruction-finetuned language models," *Journal of Machine Learning Research*, vol. 25, no. 70, pp. 1–53, 2024.

[10] D. Ghosal, N. Majumder, A. Mehrish, and S. Poria, "Text-to-audio generation using instruction-tuned llm and latent diffusion model," *arXiv preprint arXiv:2304.13731*, 2023.

[11] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," *arXiv preprint arXiv:2210.13438*, 2022.

[12] R. Yamamoto, E. Song, and J.-M. Kim, "Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6199–6203.

[13] A. Gritsenko, T. Salimans, R. van den Berg, J. Snoek, and N. Kalchbrenner, "A spectral energy distance for parallel speech synthesis," *Advances in Neural Information Processing Systems*, vol. 33, pp. 13 062–13 072, 2020.

[14] F. Kreuk, G. Synnaeve, A. Polyak, U. Singer, A. Défossez, J. Copet, D. Parikh, Y. Taigman, and Y. Adi, "Audiogen: Textually guided audio generation," *arXiv preprint arXiv:2209.15352*, 2022.

[15] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[16] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9729–9738.

[17] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of machine learning research*, vol. 21, no. 140, pp. 1–67, 2020.

[18] Q. Kong, Y. Cao, H. Liu, K. Choi, and Y. Wang, "Decoupling magnitude and phase estimation with deep resunet for music source separation," *arXiv preprint arXiv:2109.05418*, 2021.

[19] X. Mei, C. Meng, H. Liu, Q. Kong, T. Ko, C. Zhao, M. D. Plumbley, Y. Zou, and W. Wang, "Wavcaps: A chatgpt-assisted weakly-labelled audio captioning dataset for audio-language multimodal research," *arXiv preprint arXiv:2303.17395*, 2023.

[20] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[21] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "Sdr–half-baked or well done?" in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 626–630.

DCASE 2024 TASK 4:
SOUND EVENT DETECTION WITH HETEROGENEOUS DATA AND MISSING LABELS

Samuele Cornell[1,*], *Janek Ebbers*[2,*], *Constance Douwes*[3],
Irene Martín-Morató[4], *Manu Harju*[4], *Annamaria Mesaros*[4], *Romain Serizel*[3]

[1]Carnegie Mellon University, USA [2]Mitsubishi Electric Research Laboratories, USA
[3] Universite de Lorraine, CNRS, Inria, Loria, Nancy, France [4] Tampere University, Finland

ABSTRACT

The Detection and Classification of Acoustic Scenes and Events Challenge Task 4 aims to advance sound event detection (SED) systems by leveraging training data with different supervision uncertainty. Participants are challenged in exploring how to best use training data from different domains and with varying annotation granularity (strong/weak temporal resolution, soft/hard labels), to obtain a robust SED system that can generalize across different scenarios. Crucially, annotation across available training datasets can be inconsistent and hence sound events of one dataset may be present but not annotated in an other one. As such, systems have to cope with potentially missing target labels during training. Moreover, as an additional novelty, systems are also evaluated on labels with different granularity in order to assess their robustness for different applications. To lower the entry barrier for participants, we developed an updated baseline system with several caveats to address these aforementioned problems. Results with our baseline system indicate that this research direction is promising and it is possible to obtain a stronger SED system by using diverse domain training data with missing labels compared to training a SED system for each domain separately.

Index Terms— Sound event detection, missing labels, efficiency, weak supervision, heterogeneous data

1. INTRODUCTION

It can be argued that, with current deep learning based techniques, the ability to leverage as much training data as possible is as important as the pursue of novel (in the methodological sense) techniques [1]. For example, the effectiveness of modern large-language models (LLMs) relies mostly on the scale of the training data rather than on their deep neural network (DNN) architecture. The same is true for automatic speech recognition (ASR) models, with recent works [2–4] demonstrating that a great deal of robustness, as well as zero-shot and emerging capabilities [2], come both from the scale of the model and, crucially, the size of the training set.

However, leveraging data at scale has its own set of challenges. This is particularly true for SED where readily available data and metadata is not effortlessly obtainable from web sources. While self-supervised learning (SSL) techniques [5–8] can help to circumvent this issue, supervised data is still necessary for fine-tuning. For this latter, the only viable option right now is manual annotation, which is very expensive and difficult to scale as SED requires temporal endpoints together with the class label. To lower the annotation burden, temporally *weak* annotations (i.e. presence or not of a sound event inside a particular audio clip of several seconds without precise endpoints) are often used in conjunction with a smaller portion of temporally precise (i.e. *strong*) annotated recordings [9, 10]. These latter are particularly important, as it has been demonstrated [11,12] that increasing the amount of strongly-labeled examples brings considerable benefits in terms of performance, despite the obvious drawbacks of increasing the annotations costs. As such, in the recently proposed MAESTRO [13] dataset, a sliding window approach to the annotation procedure was developed. This approach, together with crowdsourcing, allows for better scaling in the annotation stage. In MAESTRO, temporally strong labels are obtained by overlap-add of several temporally weak annotations.

This discrepancy in the annotation temporal granularity has been explored extensively in the past DCASE Task 4 challenges [10, 14–19] since 2018, with DESED [20, 21] being the main dataset used through all these past editions.

However, another crucial issue is that, between different datasets, not only the temporal granularity (temporally strong vs. weak labels) can vary but also the consistency in the annotation procedure, i.e. which classes are considered as events of interest and which are instead disregarded, or again, if annotation confidence (i.e. the use of *soft* labels) is available or not. This direction has been largely underexplored in previous DCASE Task 4 challenges but is essential towards the goal of leveraging as much as training data as possible and is the main novelty introduced this year.

2. MOTIVATION

This year the DCASE Challenge Task 4 aims at addressing two different aspects related to the aforementioned problem of leveraging diverse training data with missing and (temporally and/or posterior-wise) weak annotation. Each of these aspects answer fundamental research questions which are formulated in the following.

2.1. Can we combine datasets from diverse domains with different annotations to improve performance ?

One of the many challenges of combining different datasets for SED is the fact that datasets may not have consistent annotation with one another. In extreme cases, the datasets might not even share any common sound event classes. Instead of training a SED model on each dataset separately an intriguing approach is to just train one model on all available datasets. Intuitively, if two datasets have sound classes that overlap or, at least, some classes that could be mapped from one another (e.g. when one event is a sub-class of another event [22–24]), then we expect that using both datasets should afford better performance compared to training a model for each separately. However, since annotation can be inconsistent and some events that are annotated in one dataset may be present but not annotated in the other, the training procedure and possibly even the SED model must be modified to account for this issue. In Section 6 we

[*] These authors contributed equally to this work

describe how we addressed this when developing this year baseline system and in Section 7.2 we present some results which indicate that this research direction is promising and indeed leads to large performance gains.

2.2. What is the best way to exploit soft labels ? Are they useful to improve performance ?

Some datasets, such as MAESTRO, due to their data annotation protocol, have soft labels expressing the annotators overall confidence of the presence or not of a particular sound event. In [13] it was shown that it is possible to train an effective SED system using such soft labeled annotation and two possible loss functions: binary cross entropy (BCE) and mean square error (MSE) were explored, as well as different post-processing techniques. In particular, the choice of the loss function was found to affect the model performance on more rarely occurring sound event classes. Several research questions however arise when soft labels are combined with strong labels from other datasets and with soft labels from pseudo labels obtained from the model (e.g. via mean-teacher [25]). It would be interesting to assess if annotation confidence metadata is useful for training a robust SED system when training data is scaled, and if also other approaches e.g. filtering are helpful or not.

3. CHALLENGE DATASETS

This year the challenge keeps using the DESED dataset, in order to be comparable with previous editions, but adds MAESTRO as another dataset participants can use and on which performance will be evaluated. Both are described in detail in the following.

DESED consists of 10 seconds length audio clips either recorded in a domestic environment or synthesized to reproduce such an environment. It features annotated sound events from 10 different classes: alarm_bell_ringing, blender, cat, dishes, dog, electric_shaver_toothbrush, frying, running_water, speech, vacuum_cleaner. The synthetic part of the dataset is generated with Scaper [26] with foreground events obtained from the Freesound dataset [27] while backgrounds are extracted from YouTube videos under Creative Commons license, Freesound subset of the MUSAN dataset [28] and SINS [29]. The synthetic set is divided into an evaluation and training part. More information is available in [16]. The real-world recording part is instead derived from AudioSet [30] and it comprises of a temporally-weakly annotated set (1578 clips), a totally unlabeled set (14412 clips) and also a strongly annotated portion obtained with the procedure described in [11] (3470 clips).

MAESTRO Real, which has been proposed in [13] and used in the past DCASE 2023 Task 4 (track B) challenge, consists of a development (6426 clips) and an evaluation part of long-form real-world recordings. This dataset contains multiple temporally-strong annotated events with soft labels from 17 classes. However, in this challenge, out of these, only 11 are considered in evaluation as the other 6 do not occur with confidence over 0.5. These classes are: birds_singing, car, people_talking, footsteps, children_voices, wind_blowing, brakes_squeaking, large_vehicle, cutlery_and_dishes, metro_approaching, metro_leaving. As said, this data was annotated using crowdsourcing and the procedure introduced in [31], where temporally-weak labeling is used in conjunction to a sliding window approach to derive events temporal localization. Multiple annotators outputs are aggregated via MACE [32]. The recordings are derived from TUT Acoustic Scenes 2016 [33] dataset and are between 3 to 5 minutes long.

4. RULES

Rules are largely similar to previous year edition. However this year we allow participants to use external data and pre-trained models [1]. Another important difference is that, this year, since we have two scenarios, we prohibit domain identification. In fact we want participants to focus on approaches that can generalize across various scenarios without apriori knowledge of which subset of sound classes can be present.

5. EVALUATION

SED evaluation assesses a system's capability of recognizing and temporally localizing sound events. Currently three different event-matching approaches exist, namely collar- [34], intersection- [35, 36] and segment-based [34], which differ in the way they compare predicted and ground truth temporal locations of sound events. In recent years, intersection-based evaluation has gained popularity as an event-based metric favoring detection of reasonably connected events, while being less sensitive to annotation ambiguities compared to collar-based evaluations. Further, there is a high variation in SED application requirements, with some applications requiring a high recall, others a high precision, and yet others may even let the user control sensitivity. Hence, an SED evaluation metric ideally aggregates performance over various operating modes.

Therefore, the polyphonic sound detection score (PSDS) [35, 37] has been used as primary metric in this task since 2021. It evaluates the normalized partial area under the PSD-ROC curve, where the PSD-ROC is the average of class-wise intersection-based ROC curves plus a penalty on inter-class standard deviation. PSDS parameters are the detection tolerance criterion ρ_{DTC} (the required intersection of a detected event with ground truth events to not be counted false positive (FP)), the ground truth intersection criterion ρ_{GTC} (the required intersection of a ground truth event with non-FP detected events to be counted true positive (TP)), the penalty weight α_{ST} on inter-class standard deviation, and the maximum FP-rate $e_{\max}$ up to which the area under curve is computed [2]. In previous editions PSDS1 and PSDS2 have been evaluated, which differ in their parameters. This year we are considering only PSDS1 for evaluation with $\rho_{\text{DTC}} = \rho_{\text{GTC}} = 0.7$, $\alpha_{\text{ST}} = 1.$, $e_{\max} = 100\,\text{FPs/hour}$, as PSDS2 is tuned more as an audio tagging than an SED metric. Events onset and offset times required for PSDS computation, however, are only available for DESED data and classes, which is why PSDS1 is only evaluated on this fraction of the evaluation set.

For MAESTRO, segment-based labels (segment length of one second) are provided, and we use the segment-based mean (macro-averaged) partial area under ROC curve (segMPAUC) as the primary metric instead, with a maximum FP-rate of $e_{\max} = 0.1$. To better match the PSDS calculation, we don't use McClish correction [38], but only normalize by $e_{\max}$ yielding segMPAUC $\in [\frac{e_{\max}}{2}, 1]$. segMPAUC is computed w.r.t. hard labels (using a binarization threshold of 0.5) for the 11 classes listed in Sec. 3.

To have a common processing of DESED and MAESTRO data during inference, we split MAESTRO recordings, which comprise several minutes, into clips of 10 seconds with a clip overlap of 50%. DESED and MAESTRO clips are anonymized and shuffled in the evaluation set to prevent manual domain identification (cf. task rules in Sec. 4). At evaluation time, we reconstruct recording-level

[1] Allowed data and model resources are listed in the challenge website

[2] Cross-trigger parameters are not mentioned as not considered this year.

predictions from the MAESTRO clips by computing, for each class, a scalar posterior score in each segment. To do so, submitted (short-time) class posterior scores are obtained, first by averaging over the duration of a segment and, secondly, by averaging segment-level scores of the same segment from overlapping clips.

In addition to the primary metrics ($\text{PSDS1}_{\text{DESED}}$ and $\text{segMPAUC}_{\text{MAESTRO}}$), we report segMPAUC on DESED ($\text{segMPAUC}_{\text{DESED}}$), macro-averaged collar-based F_1-scores on DESED, and macro-averaged segment-based F_1-scores on DESED and MAESTRO for a detection threshold of 0.5 and for optimal detection thresholds. All metrics are evaluated using sed_scores_eval[3]. As in previous editions, we use both the predictions from three independent training runs and bootstrapped evaluation [39] to compute metrics' means and standard deviations. For DESED, 20 different bootstrap samples (whereby we ensure that each clip is overall sampled equally often) are evaluated for each of the three runs yielding 60 results to compute statistics from. For MAESTRO, statistics are only computed over the three independent training runs as otherwise some classes may not have any positive instances in a bootstrap sample due to the small number of evaluation files. As ranking metric the sum of the primary metrics' means $\overline{\text{PSDS1}_{\text{DESED}}} + \overline{\text{segMPAUC}_{\text{MAESTRO}}}$ is used. Note that both metrics are taken from the same system, as, in contrast to previous editions, both metrics focus on SED here.

Energy efficiency is another important factor in SED systems. As in the previous two editions, we ask participants to report the energy consumption of their system during both training and testing stages using the CodeCarbon package [40]. We also ask participants to report the energy consumption for training the baseline model on 10 epochs as well as for inference with the baseline model on the development set. This procedure has to be performed on the same hardware as used for their system training/inference such that energy consumption can be normalized among different hardware and provide fairer comparisons [18]. In addition, this year we ask not only CodeCarbon's total energy consumption, which is calculated as the sum of the three components (GPU, CPU, RAM), but also the energy from the GPU component alone. In fact, we found that CPU and RAM consumption due to dataloading were included by CodeCarbon in previous DCASE Task 4 challenges, while we are also interested in an accurate picture of the GPU energy alone. Having a more precise energy consumption estimation could allow to better assess the relationship between the number of multiply-accumulate (MAC) operations, the number of parameters, and energy consumption from the GPU. Section 7, Table 1 reports energy consumption figures for the baseline.

6. DCASE 2024 CHALLENGE TASK 4 BASELINE SYSTEM

The baseline system is directly inherited from the previous 2023 DCASE Task 4 challenge [19] and consists of a convolutional recurrent neural network (CRNN) network which also employs self-supervisedly learned features from BEATs pre-trained model [7]. The CRNN model has a convolutional neural network (CNN) encoder of 7 convolutional layers with batch normalization, gated linear unit and dropout, followed by a bi-directional gated recurrent unit (biGRU) layer. Before this latter, BEATs features are concatenated with the CNN extracted ones. Average pooling is applied to BEATs features to make the sequence length the same as the one from the CNN encoder. Clip-wise and frame-wise posteriors are then derived using an attention pooling [41]. The CNN encoder is fed log-mel filterbank energies extracted with a 128 ms window and 16 ms stride from 16 kHz audio. During training the BEATs model is kept frozen, Mixup [42] regularization strategy is employed and the mean-teacher framework [25] is used in order to leverage unlabeled and weakly-labeled data. Baseline code and pre-trained checkpoints are available online[4].

For this year challenge we introduced two incremental improvements and, to deal with the aforementioned missing labels problem, also some ad-hoc modifications to the training procedure. Regarding the minor improvements, for this year baseline we use SpecAugment-style [43] time-wise masking on the features extracted by the pre-trained model and, independently, on the features extracted from the CNN encoder. We denote this strategy as *dropstep* in Section 7.1. Another difference is that for post-processing we employ a multi-class median filter where each class has a different median filter length.

6.1. Dealing with partially annotated data

The training procedure had to be modified in several places in order to deal with the missing labels problem.

1) Cross mapping sound event classes: first, as a pre-processing step, we map some DESED events to similar classes in MAESTRO. More in detail, we have in DESED "speech" which is a super-class for "people_talking, children_voices, announcement" in MAESTRO, "dishes" which corresponds to "cutlery_and_dishes" and also "dog" which is a super-class for "dog_bark". Note that these mapping are from MAESTRO to DESED but not vice-versa as DESED ones are mostly super-classes of MAESTRO ones. Intuitively, with this strategy, when computing the loss on MAESTRO e.g. for a clip with the event "people_talking" having confidence 0.5, we also drive the network output posterior corresponding to "speech" class to 0.5.

2) Loss computation: the model is trained using BCE loss function on real-world strongly, synthetic and weakly labeled examples as well as on MAESTRO soft labeled examples. MSE is instead used for the mean-teacher pseudo-labeling loss component which is applied on both weak and unlabeled data from DESED. When computing the loss for both components on a particular clip we avoid computing the loss for the network outputs corresponding to the classes that do not correspond to the clip original dataset. For example, for MAESTRO, we do not compute the loss for DESED output logits except for classes that have been cross-mapped as explained before.

3) Attention-pooling masking: the attention pooling mechanism [41] employed in the final layer of the baseline model applies the softmax function over classes. Before taking the softmax, the values corresponding to unlabeled classes (not belonging to the current clip dataset) are masked to minus infinite in order to prevent to attend to them.

4) Mixup: Mixup [42] regularization strategy is applied for MAESTRO and DESED independently as labels are missing and the two cannot be mixed together in a reliable manner.

6.2. Hyperparameters tuning

We adopt a dual-phase approach to hyperparameters tuning in order to ease the computational burden of the overall tuning procedure. In the first step, we tune the network and training parameters [5]. This

[3] https://github.com/fgnt/sed_scores_eval

[4] github.com/DCASE-REPO/DESED_task/recipes/dcase2024_task4_baseline

[5] Script available at: dcase2024_task4_baseline/optuna_pretrained.py

	300 epochs	10 epochs	Dev-test
Total Energy (kWh)	0.9458 ± 0.0708	0.0299 ± 0.0011	0.0682 ± 0.0007
GPU Energy (kWh)	0.3127 ± 0.0160 (33%)	0.0103 ± 0.0008 (34%)	0.0116 ± 0.0004 (17%)
CPU Energy (kWh)	0.2203 ± 0.0205 (23%)	0.0068 ± 0.0002 (23%)	0.0197 ± 0.0001 (29%)
RAM Energy (kWh)	0.4129 ± 0.0391 (44%)	0.0128 ± 0.0004 (43%)	0.0369 ± 0.0003 (54%)
Duration (s)	7929 ± 737	244 ± 8	708 ± 4

Table 1: Baseline energy consumption for training and inferring on the development set, both DESED and MAESTRO, on one A100 (40GB)

Model	PSDS1 ↑	segMPAUC ↑
	Dev-test (DESED)	Dev-test (MAESTRO)
Random Init	0.0	0.02
Baseline	0.491	0.731
- dropstep	0.479	0.706
- HypTune1	0.458	0.669
- HypTune2	0.391	0.702
- MC-Median	0.485	0.714
- DESED	0.0	0.642
- MAESTRO	0.483	0.115
- CrossMap	0.469	0.722

Table 2: Baseline improvements ablation study on dev-test and effect of training the system only on DESED or MAESTRO data. For MAESTRO, we used 90% overlap when reconstructing the long-form audio.

requires training the model from scratch for each set of selected hyperparameters. In detail we tune the number of biGRU layers and its hidden state size, learning rate, dropout and dropstep parameters, warmup epochs and gradient clipping value. In a second step, the network is kept frozen and we use the best model as found in the first step and tune only the multi-class median filter. This second step requires only to perform inference on the dev-test portions of the data[6]. Such dual-phase approach allows for dramatically reducing the required number of training runs compared to tuning everything together from scratch, since a slight change in the median filter length for a particular class has a significant effect on the performance of the overall system, leading to a very noisy hyperparameter tuning procedure. This procedure was performed using the Optuna toolkit [44] using multi-objective tree-structured Parzen estimator [45] with dev-test $\overline{\text{PSDS1}}_{\text{DESED}} + \overline{\text{segMPAUC}}_{\text{MAESTRO}}$ as the objective function.

7. EXPERIMENTAL RESULTS

7.1. Baseline improvements

In Table 2 top-panel, we report an ablation study to motivate the baseline system changes described in Sec. 6. We can observe that all the proposed changes bring substantial improvement. In particular, the dual-phase Optuna-based hyperparameter tuning (- HypTune ablations) appears to be quite effective. Adding a median filter (- HypTune 2 ablation, unprocessed scores) seems crucial, while having a multi-class median filter (- MC-Median ablation), improves performance only marginally. Compared to this latter, the dropstep regularization strategy has a more significant effect (- dropstep ablation).

7.2. Leveraging heterogeneous datasets with missing labels

In Table 2 bottom-panel we report an ablation study to assess how removing one of the two datasets (MAESTRO or DESED) affects the overall performance of the SED system. We can see that, in both instances where the other dataset is removed, whether it is DESED (- MAESTRO ablation) or MAESTRO (- DESED ablation), the performance on the remaining dataset also drops. However, the performance drop on DESED is small if MAESTRO is removed. This is likely due to the fact that DESED is much larger and thus the effect of removing/adding MAESTRO is modest. The strategy described in Section 6 of mapping some MAESTRO classes to some DESED classes is considerably effective (- CrossMap ablation) in particular for DESED as one would expect (some MAESTRO classes are mapped to corresponding DESED super-classes). What is rather surprising is, instead, the fact that if DESED is removed (- DESED ablation), the performance on MAESTRO drops quite dramatically. In fact, as described in Section 6, during training, when both datasets are used, the loss on the classes that do not belong to the dataset from which the input audio is taken are masked, thus e.g. MAESTRO outputs are completely ignored when the input audio comes from DESED (we do not map any class from DESED to MAESTRO). We hypothesize that the addition of DESED data boosts significantly the performance on MAESTRO because it may help the model to learn how to extract a more meaningful and generalizable representation especially in the earlier layers of the network, acting as a regularization strategy (especially important as MAESTRO is small compared to DESED). This hypothesis may also explain why if we remove the class mapping (- CrossMap ablation) the performance on MAESTRO is still superior to using MAESTRO alone.

8. CONCLUSIONS

In this paper we presented the DCASE 2024 Task 4 challenge which addresses the important problem of leveraging multiple data sources for training SED systems. Datasets can differ in the temporal resolution of the labels e.g. temporally *strong* or *weak* labels or in the fact that annotator confidence may be present (e.g. *soft* labels) or not, or again, by which sound classes are actually considered during the annotation process. To spur research towards addressing these issues, this year task involves two datasets DESED and MAESTRO on which participants systems are benchmarked, while external data and pre-trained models can also be leveraged. Due to the aforementioned annotation inconsistencies participants need to devise novel and effective ways to cope with the fact that sound events that are considered in DESED may be present in MAESTRO but are not annotated and vice versa. To ease the challenge participation entry barrier, an updated baseline system was developed. Results from such baseline suggest that leveraging more data, if the aforementioned problems are addressed in a reasonable way, is always beneficial. In fact, we show that it is possible to obtain a system trained on multiple datasets which is stronger than single systems that are trained on each dataset/scenario independently.

[6] The optimized class-wise median filters lengths are in dcase2024_task4_baseline/confs/default.yaml

9. REFERENCES

[1] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting unreasonable effectiveness of data in deep learning era," in *Proc. of ICCV*, 2017, pp. 843–852.

[2] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *International Conference on Machine Learning*, 2023, pp. 28 492–28 518.

[3] Y. Peng, J. Tian, B. Yan, D. Berrebbi, X. Chang, X. Li, J. Shi, S. Arora, W. Chen, R. Sharma, *et al.*, "Reproducing Whisper-style training using an open-source toolkit and publicly available data," in *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2023, pp. 1–8.

[4] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao, *et al.*, "WavLM: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.

[5] Y. Gong, C.-I. Lai, Y.-A. Chung, and J. Glass, "SSAST: Self-supervised audio spectrogram transformer," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 10, 2022, pp. 10 699–10 709.

[6] A. Baade, P. Peng, and D. Harwath, "MAE-AST: Masked autoencoding audio spectrogram transformer," 2022.

[7] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, Z. Chen, W. Che, X. Yu, and F. Wei, "BEATs: Audio pre-training with acoustic tokenizers," in *International Conference on Machine Learning*, 2023, pp. 5178–5193.

[8] P.-Y. Huang, H. Xu, J. Li, A. Baevski, M. Auli, W. Galuba, F. Metze, and C. Feichtenhofer, "Masked autoencoders that listen," *Advances in Neural Information Processing Systems*, vol. 35, pp. 28 708–28 720, 2022.

[9] R. Serizel, N. Turpault, H. Eghbal-Zadeh, and A. P. Shah, "Large-scale weakly labeled semi-supervised sound event detection in domestic environments," *arXiv preprint arXiv:1807.10501*, 2018.

[10] R. Serizel, N. Turpault, H. Eghbal-Zadeh, and A. Parag Shah, "Large-Scale Weakly Labeled Semi-Supervised Sound Event Detection in Domestic Environments," in *DCASE Workshop*, 2018.

[11] S. Hershey, D. P. Ellis, E. Fonseca, A. Jansen, C. Liu, R. C. Moore, and M. Plakal, "The benefit of temporally-strong labels in audio event classification," in *Proc. of ICASSP*, 2021, pp. 366–370.

[12] F. Ronchini, R. Serizel, N. Turpault, and S. Cornell, "The impact of non-target events in synthetic soundscapes for sound event detection," *DCASE Workshop*, 2021.

[13] I. Martín-Morató, M. Harju, P. Ahokas, and A. Mesaros, "Training sound event detection with soft labels from crowdsourced annotations," in *Proc. of ICASSP*, 2023, pp. 1–5.

[14] N. Turpault and R. Serizel, "Training sound event detection on a heterogeneous dataset," in *DCASE Workshop*, 2020.

[15] N. Turpault, R. Serizel, A. Parag Shah, and J. Salamon, "Sound event detection in domestic environments with weakly labeled data and soundscape synthesis," in *DCASE Workshop*, 2019.

[16] R. Serizel, N. Turpault, A. Shah, and J. Salamon, "Sound event detection in synthetic domestic environments," in *Proc. of ICASSP*, 2020.

[17] F. Ronchini, S. Cornell, and N. e. a. Turpault, "DCASE 2021 Task 4 Challenge," https://dcase.community/challenge2021, 2021.

[18] F. Ronchini, S. Cornell, R. Serizel, N. Turpault, E. Fonseca, and D. P. Ellis, "Description and analysis of novelties introduced in dcase task 4 2022 on the baseline system," *DCASE Workshop*, 2022.

[19] F. Ronchini, J. Ebbers, F. Angulo, D. Perera, S. Essid, and R. Serizel, "DCASE 2023 Task 4a Challenge," https://dcase.community/challenge2023, 2023.

[20] R. Serizel, N. Turpault, A. Shah, and J. Salamon, "Sound event detection in synthetic domestic environments," in *Proc. of ICASSP*, 2020.

[21] N. Turpault, R. Serizel, A. Parag Shah, and J. Salamon, "Sound event detection in domestic environments with weakly labeled data and soundscape synthesis," in *Detection and Classification of Acoustic Scenes and Events, Workshop, DCASE*, 2019.

[22] H. Shrivastava, Y. Yin, R. R. Shah, and R. Zimmermann, "Mt-gcn for multi-label audio-tagging with noisy labels," in *Proc. of ICASSP*, 2020, pp. 136–140.

[23] G. Wichern, B. Mechtley, A. Fink, H. Thornburg, and A. Spanias, "An ontological framework for retrieving environmental sounds using semantics and acoustic content," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2010, pp. 1–11, 2010.

[24] A. Shah, L. Tang, P. H. Chou, Y. Y. Zheng, Z. Ge, and B. Raj, "An approach to ontological learning from weak labels," in *Proc. of ICASSP*, 2023, pp. 1–5.

[25] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Advances in neural information processing systems*, 2017.

[26] J. Salamon, D. MacConnell, M. Cartwright, P. Li, and J. P. Bello, "Scaper: A library for soundscape synthesis and augmentation," in *Proc. of WASPAA*, 2017.

[27] E. Fonseca, J. Pons Puig, X. Favory, F. Font Corbera, D. Bogdanov, A. Ferraro, S. Oramas, A. Porter, and X. Serra, "Freesound datasets: a platform for the creation of open audio datasets," in *Proceedings of the 18th ISMIR Conference*, 2017.

[28] D. Snyder, G. Chen, and D. Povey, "MUSAN: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[29] G. Dekkers, S. Lauwereins, and T. et al., "The SINS database for detection of daily activities in a home environment using an acoustic sensor network," in *DCASE Workshop*, 2017.

[30] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audioset: An ontology and human-labeled dataset for audio events," in *Proc. of ICASSP*, 2017, pp. 776–780.

[31] I. Martín-Morató and A. Mesaros, "Strong labeling of sound events using crowdsourced weak labels and annotator competence estimation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 902–914, 2023.

[32] D. Hovy, T. Berg-Kirkpatrick, A. Vaswani, and E. Hovy, "Learning whom to trust with MACE," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2013, pp. 1120–1130.

[33] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *EUSIPCO*, 2016.

[34] ——, "Metrics for polyphonic sound event detection," *Applied Sciences*, 2016.

[35] Ç. Bilen, G. Ferroni, and F. e. a. Tuveri, "A framework for the robust evaluation of sound event detection," in *Proc. of ICASSP*, 2020.

[36] G. Ferroni, N. Turpault, J. Azcarreta, F. Tuveri, R. Serizel, Ç. Bilen, and S. Krstulović, "Improving sound event detection metrics: insights from dcase 2020," in *Proc. of ICASSP*, 2021, pp. 631–635.

[37] J. Ebbers, R. Haeb-Umbach, and R. Serizel, "Threshold independent evaluation of sound event detection scores," in *Proc. of ICASSP*, 2022, pp. 1021–1025.

[38] D. K. McClish, "Analyzing a portion of the roc curve," *Medical decision making*, vol. 9, no. 3, pp. 190–195, 1989.

[39] B. Efron and R. J. Tibshirani, *An introduction to the bootstrap*. Chapman and Hall/CRC, 1994.

[40] V. Schmidt, K. Goyal, A. Joshi, B. Feld, L. Conell, N. Laskaris, D. Blank, J. Wilson, S. Friedler, and S. Luccioni, "CodeCarbon: Estimate and Track Carbon Emissions from Machine Learning Computing," 2021.

[41] L. JiaKai, "Mean teacher convolution system for DCASE 2018 Task 4," DCASE2018 Challenge, Tech. Rep., 2018.

[42] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations*, 2018.

[43] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[44] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 2623–2631.

[45] Y. Ozaki, Y. Tanigaki, S. Watanabe, M. Nomura, and M. Onishi, "Multiobjective tree-structured parzen estimator," *Journal of Artificial Intelligence Research*, vol. 73, pp. 1209–1250, 2022.

FREQUENCY TRACKING FEATURES FOR DATA-EFFICIENT DEEP SIREN IDENTIFICATION

Stefano Damiano[1*], *Thomas Dietzen*[1], *Toon van Waterschoot*[1*],

[1] KU Leuven, Dept. of Electrical Engineering (ESAT-STADIUS), Leuven, Belgium,
{stefano.damiano, thomas.dietzen, toon.vanwaterschoot}@esat.kuleuven.be

ABSTRACT

The identification of siren sounds in urban soundscapes is a crucial safety aspect for smart vehicles and has been widely addressed by means of neural networks that ensure robustness to both the diversity of siren signals and the strong and unstructured background noise characterizing traffic. Convolutional neural networks analyzing spectrogram features of incoming signals achieve state-of-the-art performance when enough training data capturing the diversity of the target acoustic scenes is available. In practice, data is usually limited and algorithms should be robust to adapt to unseen acoustic conditions without requiring extensive datasets for re-training. In this work, given the harmonic nature of siren signals, characterized by a periodically evolving fundamental frequency, we propose a low-complexity feature extraction method based on frequency tracking using a single-parameter adaptive notch filter. The features are then used to design a small-scale convolutional network suitable for training with limited data. The evaluation results indicate that the proposed model consistently outperforms the traditional spectrogram-based model when limited training data is available, achieves better cross-domain generalization and has a smaller size.

Index Terms— siren detection, frequency tracking, data-efficient learning, convolutional neural network

1. INTRODUCTION

The increasing level of automation of road vehicles requires robust systems that enable cars to understand their surroundings and either provide feedback to human drivers or autonomously interact with other road users. Environmental awareness is obtained by collecting information using multi-modal sensors including cameras, radar, lidar and acoustic sensors [1]. With the rich urban soundscape containing information on events happening on a road, sound detection has been widely explored for both monitoring purposes [2, 3] and to identify emergency or harmful situations that require attention [4, 5, 6, 7, 8, 9, 10, 11]. In particular, emergency vehicles (EV) are usually announced by the sound of their siren that can often be detected from a distance, before they become visible to the driver or can be identified using other sensing modalities (e.g., when obstacles occlude the line of sight or the EV is behind a corner).

Several siren identification algorithms have been proposed, with deep learning models achieving state-of-the-art performance thanks to their robustness to the diversity of siren signals (three classes of sirens exist, namely two-tone, wail and yelp, and a large variability can be observed even between sirens of the same type) and to the prominent and non-stationary traffic background noise [4, 7, 12, 13, 14, 15]. Most state-of-the-art solutions rely on a spectrogram-based time-frequency representation of sound signals fed to 2D convolutional neural networks (CNN) [13, 12, 15]. These vision-inspired architectures process the spectrogram as a 2D image and achieve high accuracy when (diverse) enough training data is at disposal. Siren identification systems are faced with several use-case specific challenges. First, models to be deployed on-vehicle should have a low complexity to run on resource-constrained embedded devices. Second, models should have a vast generalization ability to face the diverse urban soundscape: not only the background noise can significantly differ based on factors such as the landscape (e.g., urban vs. rural), the region, or the time of the day (and day of the year), but also the characteristics of siren sounds can strongly vary among different countries. Finally, in practice, the amount of available data can be limited and datasets are unlikely to capture the diversity of the target scenes. In [15], the generalization ability of state-of-the-art siren identification networks is investigated, showing that models trained on one dataset do not always generalize well to unseen domains (cross-dataset setting): using synthetic data for training purposes is thus proposed to enhance data diversity. In [14], instead, data-efficient learning is achieved by fine-tuning a pre-trained environmental audio classification model in a *few-shot* setting to identify a specific type of two-tone siren.

Aiming for data-efficiency and low complexity, in this work, we propose novel features for siren identification based on frequency tracking. In contrast to the unstructured nature of traffic noise, sirens are artificial signals generated with a simple process: all types of sirens have a harmonic behavior characterized by a periodically evolving fundamental frequency, that can be tracked over time by means of an adaptive notch filter (ANF) [16, 17]. Adopting the single-parameter ANF design proposed in [17] (KalmANF), we design a CNN model using two features, namely the tracked fundamental frequency and the power ratio between the tracked sinusoidal component, extracted by the ANF, and the full audio signal. This allows to drastically reduce the input feature size compared to using the full spectrogram, and to thus adopt low-complexity networks. In the experimental evaluation, we show that the proposed model is suitable for training with a limited amount of data, consistently outperforming a spectrogram-based CNN [13] when small training sets are used. Moreover, the proposed model is 7 times smaller than the baseline [13] and achieves improved performance in a cross-dataset setting. Accompanying code is available at [18].

[*] This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 956962, from KU Leuven internal funds C3/23/056, and from FWO Research Project G0A0424N. This paper reflects only the authors' views and the Union is not liable for any use that may be made of the contained information. The resources and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation - Flanders (FWO) and the Flemish Government.

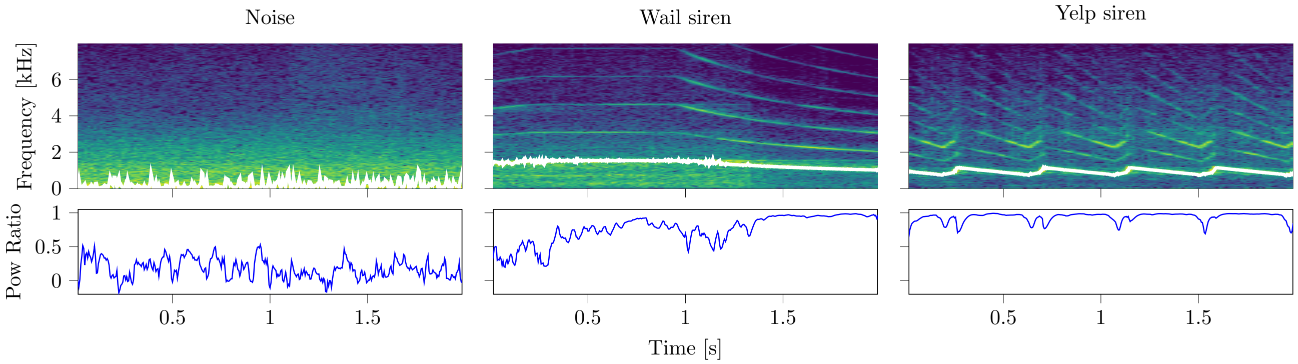


Figure 1: Proposed features for three audio samples: frequency tracked by the ANF algorithm (above, highlighted in white and overlaid to the full spectrogram) and power ratio (below).

Layer	Kernel Size	Filters/Neurons
Conv1D, stride 2	16	10
MaxPool 2x1	-	-
Conv1D, stride 2	8	20
MaxPool 2x1	-	-
Conv1D, stride 2	4	40
GlobAvgPool	-	-
Fully Connected	-	40
Fully Connected	-	20
Output	-	1

Table 1: Proposed ANFNet architecture, taking as input the two frequency tracking features.

2. PROBLEM STATEMENT AND BASELINE

We cast siren identification as a binary classification problem, where the goal is to assign a unique label (*siren* or *noise*) to a 2 s audio segment. The task is solved using the proposed architecture (ANFNet), introduced in Sec. 3, that we compare with the spectrogram-based baseline [13], denoted as VGGSiren. The network is a VGG-inspired [19] 2D-CNN composed of three blocks, each containing two 2D convolutional layers and a max pooling operation, followed by a 10-neurons FC layer and the single-neuron output layer. The network takes as input the mel-spectrogram of a 2 s-long single-channel audio segment.

3. PROPOSED METHOD

In this section, we summarize the KalmANF frequency tracking algorithm described in [17], underlining the modifications introduced to obtain the proposed features; we then present the proposed ANFNet siren identification network.

An ANF is a type of notch filter [20] whose notch frequency is recursively updated in order to suppress a high-energy sinusoidal component while leaving nearby frequencies relatively unaffected. The KalmanANF in [17] is expressed as a time-varying single-parameter bi-quadratic infinite impulse response (IIR) filter

$$H(q^{-1}, n) = \frac{1 - a(n)q^{-1} + q^{-2}}{1 - \rho a(n)q^{-1} + \rho^2 q^{-2}}, \qquad (1)$$

where n is the time index, q denotes the discrete-time shift operator defined such that, for an input signal $y(n)$, $q^{-k}y(n) = y(n-k)$ [21]; $\rho < 1$ is a fixed hyperparameter denoting the radius of the complex conjugate pole pair and $a(n) = 2\cos[2\pi f(n)/f_s]$ is the single filter parameter, $f(n)$ being the notch frequency and f_s the sampling frequency. In the KalmANF, the time-varying coefficient $a(n)$ represents the state that is adaptively estimated in order to track the variations of $f(n)$ over time, as outlined in the following. The given N-samples long input signal $y(n)$ is filtered by the direct-form II [22] of $H(q^{-1}, n-1)$ using a joint delay line for the feedforward and the feedback paths of the IIR filter. The delay line signal $s(n)$ is defined as [17]

$$s(n) = y(n) + \rho a(n-1)s(n-1) - \rho^2 s(n-2). \qquad (2)$$

In the KalmANF, $s(n)$ represents the measurement. This results in the state space model (see [17] for the detailed derivation)

$$\begin{bmatrix} a(n) \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a(n-1) \\ 1 \end{bmatrix} + \begin{bmatrix} w(n) \\ 0 \end{bmatrix} \qquad (3)$$

$$s(n) = \begin{bmatrix} s(n-1) & -s(n-2) \end{bmatrix} \begin{bmatrix} a(n) \\ 1 \end{bmatrix} + e(n), \qquad (4)$$

where $e(n)$ is the residual signal obtained at the output of the notch filter and $w(n)$ is the process noise. Based on this state-space model, $a(n)$ can be estimated in a recursive manner using the Kalman filter [23]. The estimation procedure consists in the recursive update of the covariance of the prediction error $\hat{p}(n)$, the Kalman gain $k(n)$, and the parameter estimate $\hat{a}(n)$. These steps involve scalar operations and require a memory of 2 past samples. The filter relies on tuning three hyperparameters, namely the pole radius ρ, the variance σ_e of the residual $e(n)$ and the variance σ_w of the process noise $w(n)$. First, given the previous estimate $\hat{a}(n-1)$, the measurement $s(n)$ is computed according to (2). Then, the covariance of the prediction error is computed as

$$\hat{p}(n|n-1) = \hat{p}(n-1) + \sigma_w \qquad (5)$$

and is used to obtain the Kalman gain

$$k(n) = \frac{s(n-1)}{s^2(n-1) + \frac{\sigma_e}{\hat{p}(n|n-1)}}. \qquad (6)$$

The update equation to estimate the current value of the parameter $\hat{a}(n)$ can then be expressed as

$$\hat{a}(n) = \hat{a}(n-1) + k(n)e(n), \qquad (7)$$

where, from eq. (4), the residual takes the value

$$e(n) = s(n) - \hat{a}(n-1)s(n-1) + s(n-2)\,. \tag{8}$$

Finally, the covariance of the prediction error is updated by

$$\hat{p}(n) = \left(1 - \frac{s^2(n-1)}{s^2(n-1) + \frac{\sigma_e}{\hat{p}(n|n-1)}}\right)\hat{p}(n|n-1)\,. \tag{9}$$

At each time step, the estimated parameter $\hat{a}(n)$ contains information on the frequency tracked by the ANF, that is retrieved as $\hat{f}(n) = (f_s/2\pi)\arccos\left[\hat{a}(n)/2\right]$ and will be used as a first feature for the siren identification network. It is important to notice that the tracked frequency is not necessarily the fundamental frequency, but the one with the highest energy. This comes with the advantage that, if the fundamental of a siren is missing or hidden in the background noise, the higher harmonics could still be tracked by the ANF.

We then expand the above formulation to introduce a second feature that we call *power ratio*, expressing the ratio between the power of the suppressed sinusoidal component and of the input signal. At each time step, the power of the input signal, the notched signal (after $\hat{f}$ has been suppressed) and the suppressed frequency component can be estimated recursively as

$$P_y(n) = \lambda P_y(n-1) + (1-\lambda)y^2(n) \tag{10}$$

$$P_e(n) = \lambda P_e(n-1) + (1-\lambda)e^2(n)\,, \tag{11}$$

$$P_f(n) = P_y(n) - P_e(n)\,, \tag{12}$$

where $\lambda = e^{-1/(\tau f_s)}$, with τ constituting a first additional hyperparameter representing the time constant for recursive averaging. The power ratio is finally computed as

$$P_{\text{ratio}}(n) = P_f(n)/P_y(n)\,. \tag{13}$$

To reduce the feature size, we finally downsample P_f and $\hat{f}$ by a factor q_{down}, the second additional hyperparameter of the proposed method. The procedure is summarized in Algorithm 1.

In Fig. 1 the $\hat{f}$ and P_{ratio} features are shown for a noise sample and two different siren samples (wail and yelp) extracted from the sireNNet dataset [24], that will be used for the experimental evaluation. In the top row, the tracked $\hat{f}$ is overlaid to the full spectrogram: nevertheless, we remark that the frequency estimate is obtained directly from the time-domain signal without computing the spectrogram. The visualization shows the effectiveness of the tracking algorithm when applied to siren signals, and underlines the clear contrast between the features extracted from a (structured) siren and the (unstructured) traffic noise.

We solve the siren identification problem using the ANFNet network, that processes the $\hat{f}$ and P_{ratio} features extracted from a single channel, 2 s-long audio sample: the two features are stacked into a 2-channel vector provided as input to the first layer. The architecture (see Tab. 1) contains three 1D convolutional layers (Conv1D) with, respectively, 10, 20 and 40 filters having kernel size 16, 8 and 4. Each of the first two Conv1D layers is followed by a max pooling operation (MaxPool) for dimensionality reduction. After the third one, a global average pooling operation (GlobAvgPool) is used as interface between the convolutional part and the classification head, composed of two fully connected (FC) layers with 40 and 20 neurons, respectively, and a single neuron output layer. We use the ReLU activation function in each hidden layer and the sigmoid activation in the output layer, and introduce dropout layers with 0.25 drop probability after each FC layer to prevent overfitting. The network has 7.7 k floating-point 32-bit parameters.

Algorithm 1: The modified KalmANF algorithm

```
Initialize $s(0), s(1), \hat{a}(1), \hat{p}(1) = 0$
Set $\sigma_e, \sigma_w, \rho, \tau, q_{\text{down}}$
for $n = 2$ to $N-1$ do
    $\hat{p}(n|n-1) = \hat{p}(n-1) + \sigma_w$
    $s(n) = y(n) + \rho\hat{a}(n-1)s(n-1) - \rho^2 s(n-2)$
    $k(n) = \frac{s(n-1)}{s^2(n-1) + \frac{\sigma_e}{\hat{p}(n|n-1)}}$
    $e(n) = s(n) - \hat{a}(n-1)s(n-1) + s(n-2)$
    $\hat{a}(n) = \hat{a}(n-1) + k(n)e(n)$
    $\hat{p}(n) = \left(1 - \frac{s^2(n-1)}{s^2(n-1) + \frac{\sigma_e}{\hat{p}(n|n-1)}}\right)\hat{p}(n|n-1)$
    if $|\hat{a}(n)| > 2$ then
        $\hat{a}(n) = 2\text{sgn}(\hat{a}(n))$
    end
    $\hat{f}(n) = (f_s/2\pi)\arccos\left[\hat{a}(n)/2\right]$
    $P_y(n) = \lambda P_y(n-1) + (1-\lambda)y^2(n)$
    $P_e(n) = \lambda P_e(n-1) + (1-\lambda)e^2(n)$
    $P_f(n) = P_y(n) - P_e(n)$
    $P_{\text{ratio}}(n) = P_f(n)/P_y(n)$
end
Downsample $\hat{f}$ and $P_{\text{ratio}}$ by factor $q_{\text{down}}$
```

4. EVALUATION

We run an evaluation campaign to assess the effectiveness of the proposed method: to promote reproducibility, the code is available at [18]. For training, we use the sireNNet dataset [24], containing a total of 421 noise and 1254 siren samples including different types of sirens. All samples have a duration of 3 s, and since half of the siren files are artificially generated for data augmentation purposes, we exclude them and use only the 627 non-augmented siren samples. We divide this dataset into training, validation and test data with ratios $[0.8, 0.1, 0.1]$. In order to perform a data-efficient evaluation, we split the training set into subsets of different size similarly to [25]: in particular, we create subsets containing an increasing percentage of the full training set, with ratios $0.25\%, 0.5\%, 1\%, 2\%, 4\%, 8\%, 16\%, 32\%, 64\%$ and 100% (i.e., the entire training set). 10 folds are randomly generated for each subset, in order to compute the mean and standard deviation of the results. The subsets are created such that (i) smaller splits are subsets of larger ones; (ii) the data distribution is kept similar to that of the entire training set; (iii) overlapping folds are allowed. The validation and test sets are always used without additional splitting. To further evaluate the generalization performance in a cross-dataset setting, we also use a subset of 210 audio files randomly extracted from the dataset [26] (that we will call LSSiren) for testing; this dataset contains siren and noise files with lengths between 3 s and 15 s. All files of both datasets have been re-sampled to 16 kHz and converted to mono; moreover, since we use 2 s samples as input, we take only the first two seconds of each file of the sireNNet dataset, and divide the LSSiren files in non-overlapping 2 s segments. Both datasets include real recordings, with background traffic noise, moving sirens and Doppler effect.

We implement the proposed ANFNet and the baseline VGGSiren using Pytorch Lightning [27]: for the KalmANF algorithm, we set the hyperparameters $\rho = 0.99, \sigma_w = 10^{-5}, \sigma_e = 0.66, q_{\text{down}} = 5, \tau = 0.02$, all chosen by manual tuning based on the best loss obtained on the validation set. For VGGSiren, to com-

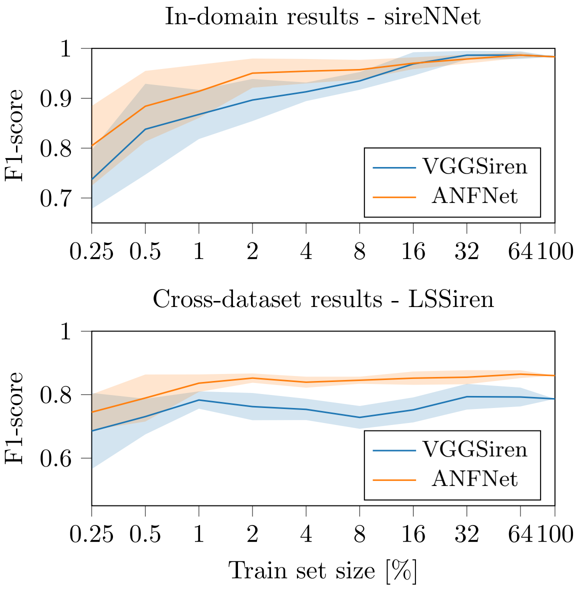


Figure 2: Comparison of the average (solid line) and standard deviation (shaded area) of the F1-score for the baseline VGGSiren and the proposed ANFNet, trained with an increasing amount of data: in-domain evaluation (above) and cross-dataset evaluation (below).

pute the mel-spectrogram we use a 1024 samples Hann window with 512 samples overlap, and 128 mel channels. As a result, VGGSiren has a total of 53.9 k floating point 32-bits parameters, thus being 7 times larger than the proposed ANFNet. For VGGSiren, we apply peak normalization to the mel-spectrograms, whereas for ANFNet we normalize the $\hat{f}$ feature to $f_s/2$ (the P_{ratio} feature is normalized by definition). In all experiments we train both models for 400 epochs using the binary cross-entropy loss function, the Adam optimizer [28] with learning rate between 0.001 and 0.005, a batch size between 2 and 32, both depending on the size of the training split, and select the best model based on the validation loss. To evaluate the performance, we use the F1-score [29] and the area under the precision-recall curve (AUPRC) [29, 30] metrics, chosen to deal with non-balanced datasets.

We train both models on the 10 folds of each sireNNet subset. Note that the 0.25%, 0.5% and 1% splits contain, respectively, only 2, 4 and 9 samples, making the problem extremely challenging and comparable to that of few-shot learning (without pre-training). First, we evaluate in-domain performance on the sireNNet test set and report in Fig. 2 the average and standard deviation (shaded area) of the F1-score. In Tab. 2 we report the average F1-score and AUPRC obtained with the two models for each training split. As expected, the performance of both networks degrades as the amount of training data decreases; nevertheless, ANFNet outperforms the baseline when trained using smaller subsets, and reaches a comparable performance on the larger ones (with a lower complexity).

We then evaluate the models on the LSSiren data (cross-dataset setting) and report the results in the bottom plot of Fig. 2 and in Tab. 3. Again, the performance of both decreases as the training dataset size decreases. In this case, the proposed ANFNet significantly outperforms the baseline on all subsets. These results indicate that the proposed features help the network capture the difference between siren and noise classes also when limited data is available, suggesting their potential for data-efficient learning. Moreover, the evaluation underlines that the proposed features ensure an enhanced robustness to domain shift compared to the mel-spectrogram. In Fig. 2 it is also visible that the standard deviation is reduced compared to VGGSiren, showing that ANFNet is less sensitive to the choice of training samples. Finally, ANFNet has a lower complexity, with a 7 times smaller network size (7.7 k parameters vs. the 53.9 k of VGGSiren). Note that, thanks to time-domain processing and downsampling, the feature extraction procedure has also a reduced complexity and the ANFNet input features have a smaller size compared to the mel-spectrograms used by VGGSiren.

	F1-score		**AUPRC**	
	VGGSiren	ANFNet	VGGSiren	ANFNet
0.25	0.7372	**0.8047**	0.8120	**0.8471**
0.5	0.8379	**0.8840**	0.8844	**0.9162**
1	0.8676	**0.9139**	0.9602	**0.9658**
2	0.8965	**0.9504**	0.9745	**0.9787**
4	0.9130	**0.9543**	**0.9781**	0.9772
8	0.9348	**0.9572**	**0.9860**	0.9796
16	0.9688	**0.9702**	**0.9949**	0.9904
32	**0.9864**	0.9787	**0.9990**	0.9962
64	**0.9865**	**0.9865**	**0.9996**	0.9966
100	**0.9833**	0.9831	**0.9995**	**0.9995**

Table 2: In-domain evaluation: average F1-score and AUPRC metrics computed on the sireNNet test set for VGGSiren and ANFNet.

	F1-score		**AUPRC**	
	VGGSiren	ANFNet	VGGSiren	ANFNet
0.25	0.6856	**0.7448**	0.6962	**0.7364**
0.5	0.7309	**0.7895**	0.7512	**0.8447**
1	0.7831	**0.8362**	0.8607	**0.9169**
2	0.7624	**0.8522**	0.8426	**0.9272**
4	0.7536	**0.8393**	0.8349	**0.9147**
8	0.7284	**0.8455**	0.7914	**0.9180**
16	0.7520	**0.8520**	0.8100	**0.9247**
32	0.7936	**0.8550**	0.8492	**0.9302**
64	0.7926	**0.8646**	0.8229	**0.9355**
100	0.7867	**0.8601**	0.8052	**0.9384**

Table 3: Cross-dataset evaluation: average F1-score and AUPRC metrics computed on LSSiren data for VGGSiren and ANFNet.

5. CONCLUSIONS

In this work, we investigated two novel features based on frequency tracking for training a siren identification model. Given the harmonic nature of siren signals, as opposed to the unstructured background noise, the features are effective for learning in a data-efficient setting, when limited data is available. The proposed system outperforms a spectrogram-based baseline on in-domain test data, when limited training data is available, and always achieves better performance in a cross-dataset setting. Moreover, its reduced complexity promotes its adoption in the automotive domain. Future work will focus on extending the frequency tracker to include higher harmonics, further investigating the generalization performance of the proposed system and optimizing the model for complexity.

6. REFERENCES

[1] L. Marchegiani and X. Fafoutis, "How Well Can Driverless Vehicles Hear? A Gentle Introduction to Auditory Perception for Autonomous and Smart Vehicles," *IEEE Intell. Transp. Syst. Mag.*, pp. 92–105, 2022.

[2] M. Won, "Intelligent Traffic Monitoring Systems for Vehicle Classification: A Survey," *IEEE Access*, vol. 8, pp. 73 340–73 358, 2020.

[3] S. Damiano, L. Bondi, S. Ghaffarzadegan, A. Guntoro, and T. van Waterschoot, "Can synthetic data boost the training of deep acoustic vehicle counting networks?" in *Proc. 2024 Int. Conf. Acoust. Speech Sig. Process. (ICASSP)*, Seoul, South Korea, 2024, pp. 631–635.

[4] F. Walden, S. Dasgupta, M. Rahman, and M. Islam, "Improving the Environmental Perception of Autonomous Vehicles using Deep Learning-based Audio Classification," *arXiv:2209.04075*, Sept. 2022.

[5] J. Yin, S. Damiano, M. Verhelst, T. van Waterschoot, and A. Guntoro, "Real-Time Acoustic Perception for Automotive Applications," in *2023 Design, Automation & Test in Europe Conf. Exhib. (DATE)*, Antwerp, Belgium, Apr. 2023, pp. 1–6.

[6] Y. Furletov, V. Willert, and J. Adamy, "Auditory Scene Understanding for Autonomous Driving," in *2021 IEEE Intell. Vehicles Symp. (IV)*, Nagoya, Japan, July 2021, pp. 697–702.

[7] L. Marchegiani and P. Newman, "Listening for Sirens: Locating and Classifying Acoustic Alarms in City Scenes," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 17 087–17 096, 2022.

[8] M. K. Nandwana and T. Hasan, "Towards Smart-Cars That Can Listen: Abnormal Acoustic Event Detection on the Road," in *Proc. 2016 Interspeech*, San Francisco, USA, Sept. 2016, pp. 2968–2971.

[9] D. Lee, S. Lee, Y. Han, and K. Lee, "Ensemble of Convolutional Neural Networks for Weakly-Supervised Sound Event Detection Using Multiple Scale Input," in *Proc. Detection Classification Acoust. Scenes Events 2017 Workshop (DCASE2017)*, Munich, Germany, Nov. 2017.

[10] A. E. Ramirez, E. Donati, and C. Chousidis, "A siren identification system using deep learning to aid hearing-impaired people," *Engineering Applications of Artificial Intelligence*, vol. 114, p. 105000, 2022.

[11] D. Pramanick, H. Ansar, H. Kumar, S. Pranav, R. Tengshe, and B. Fatimah, "Deep learning based urban sound classification and ambulance siren detector using spectrogram," in *Proc. 12th Int. Conf. Computing Comm. Networking Technologies (ICCCNT)*, Kharagpur, India, 2021, pp. 1–6.

[12] V.-T. Tran and W.-H. Tsai, "Acoustic-Based Emergency Vehicle Detection Using Convolutional Neural Networks," *IEEE Access*, vol. 8, pp. 75 702–75 713, 2020.

[13] M. Cantarini, A. Brocanelli, L. Gabrielli, and S. Squartini, "Acoustic Features for Deep Learning-Based Models for Emergency Siren Detection: An Evaluation Study," in *2021 12th Int. Symp. Image Sig. Process. Anal. (ISPA)*, Zagreb, Croatia, Sept. 2021, pp. 47–53.

[14] M. Cantarini, L. Gabrielli, and S. Squartini, "Few-Shot Emergency Siren Detection," *Sensors*, vol. 22, no. 12, p. 4338, June 2022.

[15] S. Damiano, B. Cramer, A. Guntoro, and T. van Waterschoot, "Synthetic Data Generation Techniques for Training Deep Acoustic Siren Identification Networks," *Frontiers Sig. Process.*, vol. 4, 2024.

[16] D. Rao and Sun-Yuan Kung, "Adaptive notch filtering for the retrieval of sinusoids in noise," *IEEE Trans. Acoust. Speech Sig. Process.*, vol. 32, no. 4, pp. 791–802, Aug. 1984.

[17] R. Ali and T. van Waterschoot, "A Frequency Tracker Based on a Kalman Filter Update of a Single Parameter Adaptive Notch Filter," in *Proc. 26th Int. Conf. Digital Audio Effects (DAFx)*, Copenhagen, Denmark, Sept. 2023.

[18] S. Damiano and T. Dietzen, "An ANF-based siren identification system," Github Repository, 2024. [Online]. Available: https://github.com/steDamiano/anf-siren-identification

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv:1409.1556*, 2014.

[20] K. Hirano, S. Nishimura, and S. Mitra, "Design of Digital Notch Filters," *IEEE Trans. Commun.*, vol. 22, no. 7, pp. 964–970, July 1974.

[21] T. van Waterschoot and M. Moonen, "Fifty years of acoustic feedback control: State of the art and future challenges," *Proc. IEEE*, vol. 99, no. 2, pp. 288–327, 2011.

[22] J. Travassos-Romano and M. Bellanger, "Fast least squares adaptive notch filtering," in *Proc. Int. Conf. Acoust., Speech Sig. Process. ICASSP 1988*, New York, NY, USA, 1988, pp. 1391–1394.

[23] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, Mar. 1960.

[24] A. Shah and A. Singh, "sireNNet-Emergency Vehicle Siren Classification Dataset For Urban Applications," Mendeley Data, 2023, doi: 10.17632/j4ydzzv4kb.1.

[25] F. Schmid, P. Primus, T. Heittola, A. Mesaros, I. Martín-Morató, K. Koutini, and G. Widmer, "Data-efficient low-complexity acoustic scene classification in the dcase 2024 challenge," *arXiv:1706.10006*, 2024.

[26] M. Asif, M. Usaid, M. Rashid, T. Rajab, S. Hussain, and S. Wasi, "Large-scale audio dataset for emergency vehicle sirens and road noises," *Scientific Data*, vol. 9, no. 1, p. 599, Oct. 2022.

[27] W. Falcon and The PyTorch Lightning team, "PyTorch Lightning," Github Repository, Mar. 2019. [Online]. Available: https://github.com/Lightning-AI/lightning

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, San Diego, USA, 2015.

[29] I. J. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[30] Q. Qi, Y. Luo, Z. Xu, S. Ji, and T. Yang, "Stochastic optimization of areas under precision-recall curves with provable convergence," in *Proc. 35th Int. Conf. Neural Information Process. Syst.*, online, 2021.

BASELINE MODELS AND EVALUATION OF SOUND EVENT LOCALIZATION AND DETECTION WITH DISTANCE ESTIMATION IN DCASE2024 CHALLENGE

David Diaz-Guerra[1], *Archontis Politis*[1], *Parthasaarathy Sudarsanam*[1], *Kazuki Shimada*[2], *Daniel A. Krause*[1]
Kengo Uchida[2], *Yuichiro Koyama*[3], *Naoya Takahashi*[4], *Shusuke Takahashi*[3], *Takashi Shibuya*[2]
Yuki Mitsufuji[5,6], *Tuomas Virtanen*[1]

[1] Audio Research Group, Tampere University, Tampere, Finland
[2] Sony AI, Tokyo, Japan [3] Sony Group Corporation, Tokyo, Japan
[4] Sony AI, Zurich, Switzerland [5] Sony AI, NY, USA [6] Sony Group Corporation, NY, USA

ABSTRACT

This technical report presents the objectives, evaluation, and baseline changes for Task 3, Sound Event Localization and Detection (SELD), of the DCASE2024 Challenge. While the development and evaluation dataset, STARSS23, and the division of the task into two tracks, audio-only and audiovisual (AV), remain the same, this year introduces source distance estimation (SDE) along with detection and direction-of-arrival (DOA) estimation of target sound events. Changes in task evaluation metrics and the design and training of the baseline models due to this new SDE subtask are detailed in the report and compared with the previous iteration of the challenge. Further baseline improvements regarding the integration of video information are also presented. Overall, the design of highly effective SELD models evaluated in real scenes with a limited volume of unbalanced training data has proven challenging. The introduction of SDE makes the task even more demanding, as evidenced by the low spatially-thresholded detection scores for both audio-only and AV baselines. While distance estimation error results seem promising, this comes at the expense of lower detection and DOA estimation scores compared to the previous year's baseline models without SDE. Based on the current AV model design, video integration does not bring apparent estimation benefits compared to using only audio input, indicating that more research is required into more effective fusion strategies, model architectures, data augmentation and simulation methods, or training strategies.

Index Terms— Sound event localization and detection, sound source localization, acoustic scene analysis, microphone arrays

1. INTRODUCTION

The sound event localization and detection (SELD) task, detecting the presence of sound events of target classes of interest and tracking their activity and location over time, has seen growing interest from the time of the earliest publications [1]. A large part of the research effort in this topic has been centered around the DCASE challenge[1] and the subsequent workshop, with the task developing every year in terms of data complexity and realism [2–4].

The first three iterations of the task (2019-2021) were based on synthesized spatial recordings including real ambient noise and reverberation. The data were generated with an elaborate synthesis process based on real captured multi-room and multi-point room impulse responses that allowed synthesis of both static and moving reverberant sound events [3]. Some of the task aspects that were considered in these first three SELD challenges were continuous DOA estimation, varying signal-to-noise and direct-to-reverberant ratios, moving sound sources, non-target-class interfering directional sound events, and multiple instances of the same class occurring simultaneously. The top systems of those three first challenges excelled at addressing these problems by employing improved output representations of the SELD objectives [5–7] or advanced data augmentation strategies [8].

However, those synthetic datasets lacked some important aspects of real sound scenes, mainly that of natural temporal and spatial occurences and co-occurences that characterize real sound events and their types as the result of the scene environment and the actions and interactions of the agents in it. To advance SELD research towards that direction, the next iterations up to the current one (DCASE2022-2024) were based on a new dataset of spatial recordings of real scenes [4, 9]. Annotations of sound event activities for 13 sound classes were compiled by human listeners and combined with optical tracking data of the source positions that generated those sound events. 11 hours of such material were collected in multiple rooms of two different sites. Contrary to the fairly balanced earlier synthetic datasets, the presence of classes in the real recordings was highly unbalanced, posing new challenges for the participants. To cope with the increased difficulty of the task and the small amount of training data, participants were allowed to use external data, additional simulations of recordings and pretrained audio models. Creative use of such resources [10] together with more powerful architectures driven by attention mechanisms [11] allowed the top participants to achieve competitive results with large gains over the baselines.

Additionally, in the 2023 challenge participants were allowed to use 360° video input in addition to the typical audio input [4]; an effort to foster multimodal analysis and development towards diverse large scale training of SELD systems using video supervision. Submissions of audiovisual systems did not exhibit a clear improvement using this additional modality, with only one method achieving better results than using audio-only input. This first iteration of audiovisual SELD models demonstrated that effective integration of video information was not trivial and further research and experimentation was necessary. In this year's DCASE2024 challenge the task setup remains the same, as well as the development and evaluation dataset, but with some important differences introduced otherwise. In this report, an overview of changes in the SELD task of DCASE2024 challenge is presented in terms of task objectives, baseline models, and task evaluation.

[1] `https://dcase.community/challenge2024/`

2. DISTANCE ESTIMATION

This year, we introduce a new part of the task, namely sound distance estimation. Research on DNN-based techniques for SDE has been largely confined to the binaural format. These studies typically use a classification method, assigning the source within a very limited set of distances or positions [12, 13]. A study by Kushwaha et al. [14] investigated various loss functions for distance estimation and included an activity detection component for a scenario with a tetrahedral microphone array. Few works have explored the simultaneous estimation of distance and DOA [15–17]. Until recently, there has been no effort to combine distance estimation with event detection and localization. In [18], the authors have investigated a single task and multi-task approach to 3D SELD for the binaural format and Ambisonics. Following that paper, we include some of the solutions in this years' baseline to foster further research in this area.

To employ the distance estimation task within the 3D SELD architecture, we use the multi activity-coupled Cartesian Distance and DOA (**multi-ACCDDOA**) method as described in [18]. The method is basically an extension of the multi-ACCDOA output proposed in [19]. Compared with the former, the 3-element DOA vector is extended to include the distance estimate as well. For N tracks, C classes, and T frames, the output is defined as $y_{nct} = [a_{nct}R_{nct}, D_{nct}]$, where n, c, t indicate the output track number, target class, and time frame, $a_{nct} \in \{0, 1\}$ stands for the detection activity, $R_{nct} \in \langle -1, 1 \rangle^3$ is the DOA vector, and $D_{nct} \in \langle 0, \infty)$ corresponds to distance values. The dimensions hold the following characteristics: $\mathbf{a}, \mathbf{D} \in \mathbb{R}^{N \times C \times T}, \mathbf{R} \in \mathbb{R}^{3 \times N \times C \times T}$, and $||\mathbf{R}_{nct}|| = 1$. We model up to $N = 3$ and $C = 13$. The whole output is linear to contain the range of both DOA and distance values. The multi-ACCDDOA model is trained using Auxiliary Duplicating Permutation Invariant Training (ADPIT) as in [19]. The final loss function is defined as:

$$\mathcal{L}^{ADPIT} = \frac{1}{CT} \sum_{c}^{C} \sum_{t}^{T} \min_{\alpha \in \mathrm{Perm}[ct]} l_{\alpha,ct}^{ACCDDOA}, \tag{1}$$

$$l_{\alpha,ct}^{ACCDDOA} = \frac{1}{N} \sum_{n}^{N} \mathcal{L}(y_{\alpha,nct}, \hat{y}_{\alpha,nct}), \tag{2}$$

where $\mathcal{L}(\cdot)$ is the mean square error loss function, α is one possible track permutation and $\mathrm{Perm}[ct]$ is the set of all possible permutations.

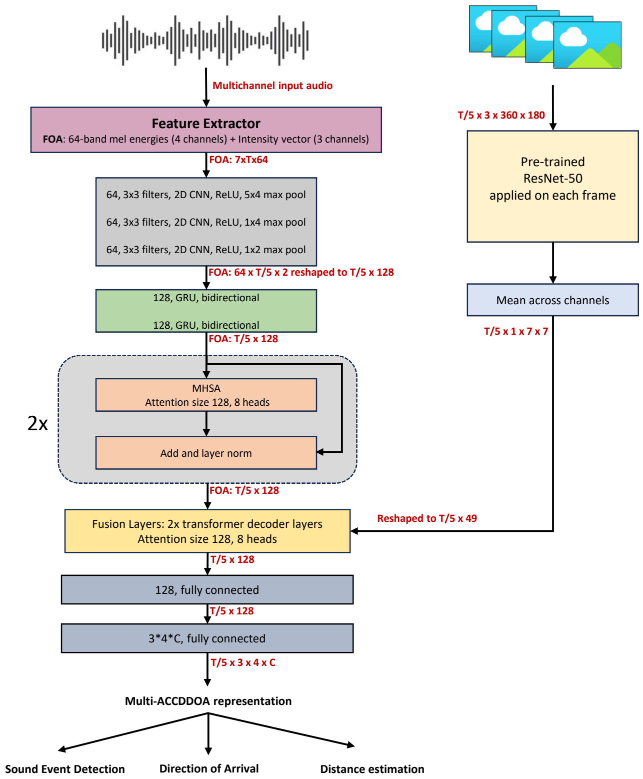


Figure 1: Audiovisual baseline model architecture.

3. BASELINE

For the audio baseline, we retain the same architecture from the previous challenge. It is a modified version of the SELDnet presented in [1]. Last year, we introduced multi-head self-attention blocks in the SELDnet architecture based on the findings in [20].

For the last year's audiovisual baseline [4], an object detector [21] was used to extract visual information. The bounding box outputs were encoded to vectors along with azimuth and elevation [22]. The encoded vectors were treated as visual features in the previous challenge [4]. In this edition of the challenge, the visual pipeline is simplified. Inspired by the work in [23], we use a pre-trained ResNet-50 [24] to extract the visual features from each frame of the video corresponding to the audio input. This visual representation of the input video is combined with the audio representation using audio-visual fusion layers. A transformer decoder block [25] with 2 layers, having an attention size of 128 with 8 heads is used for the fusion of audio and visual features. The new audio-visual baseline architecture used in the challenge is shown in Figure 1.

Differing from previous years, we changed the training procedure to fairly compare the performances of the audio-only model and the audio-visual model. In the previous iterations of the challenge, the audio baseline was trained simultaneously on the synthetic dataset and the train split of the STARSS23 development data. However, it is to be noted that the synthetic data is available only for the audio data and hence direct comparison of the audio-only and the audio-visual models was not possible. To this end, we first trained the audio baseline model on the synthetic dataset and use it for initializing the weights of the audio feature extraction layers for both the audio-only and audio-visual models. As a second step, we trained both the models on the STARSS23 development dataset. Generation of synthetic data was switched this year from the provided code by the task organizers to the more flexible spatialScaper [26] published recently.

4. EVALUATION METRICS

In previous editions of the challenge, the models were evaluated according to four metrics: localization-dependent F-score (F_{20°) and error rate (ER_{20°) and class-dependent localization error (LE_c) and localization recall (LR_c), all of them computed in one-second non-overlapping segments [2, 27]. One of our goals in this edition was to simplify the evaluation, so we decided to drop the error

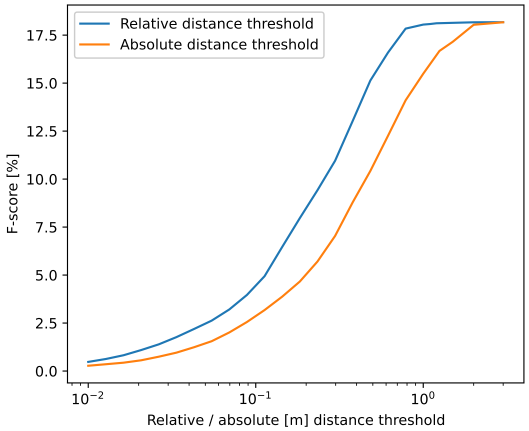


Figure 2: F-score of the 2024 audiovisual baseline system on the evaluation dataset for different values of relative and absolute thresholds. The DOA error threshold was set to 20° in all the experiments.

rate and the localization recall and keep the localization-dependent F-score (which focuses on detection) and the class-dependent localization error (which focuses on the DOA estimation) and to add a new distance estimation metric. In order to make clear that the localization error only evaluates the DOA estimation without taking into account the distance estimation, we renamed it to DOA error ($DOAE_c$).

4.1. Frame-based metrics

The computation of the metrics in one-second non-overlapping segments done in previous challenges [27] was a common practice for evaluating SED systems [28], but not for localization and tracking. It made the metrics and the evaluation code more difficult to interpret and maintain and also prevented them from being extended to more tracking-based metrics in the future, such as measuring the identity-switch ratio, which must be computed at frame level (i.e. for every time output of the system).

Therefore, we decided to compute the metrics at frame level this year. In table 1 we can see the metrics of the top-5 systems resulting from re-evaluating the audio-only systems from the previous challenge at frame level and the comparison with the original segment-based evaluation. We can see how there are no changes in the leaderboard and the metrics slightly degrade but without dramatic changes.

4.2. Distance estimation evaluation

The main novelty of this year's challenge was introducing distance estimation into the SELD task. Since we are now estimating both DOA and distance, we could have combined both into a 3D position estimation and evaluated it just as the Euclidean distance in meters to the actual source position. However, distance estimation is a more difficult task than DOA estimation when working with compact arrays due to the geometrical and physical principles of the problem, so we could expect the errors of the distance estimation to be quite larger than the ones of the DOA estimation. Hence, we preferred to keep the evaluation of both estimations separately.

Also due to the geometrical principles of the problem, distance estimation with compact arrays becomes harder when distance increases (the impact of distance in the phase differences between microphones reduces) so we decided to evaluate the distance in terms of relative distance (i.e. the ratio of the difference between the estimated and actual distance and the actual distance) instead of in absolute terms. This also fits most applications, where an absolute error of a few centimeters is more important if the source is closer to the microphones than if it is several meters away.

We did not want poor distance estimations to penalize the F-score too much this year, so we chose a relative error threshold of 1 so only really large errors have an impact on it. Figure 2 shows how the F-score of the baseline degrades when the distance estimation error threshold is reduced. In the following editions of the challenge, we will adjust the threshold according to the performance of the systems submitted this year.

4.3. Estimate-reference assignment

When we have several estimated and/or reference events of the same class simultaneously, we need to assign the estimates to the references before computing the evaluation metrics. In previous editions of the challenge, we did this by using the Hungarian algorithm [29] to find the assignment that minimized the DOA error. As previously explained, since this year we also have distance estimation, we can compute the localization error (LE) defined as the Euclidean distance between the estimate and the reference position, so we could use the Hungarian algorithm to minimize this metric instead of the DOAE. However, since we are not using this LE as an evaluation metric, we decided to maintain the estimate-reference assignment as in previous editions of the challenge.

Table 2 compares the results of the audio-only baseline model when the assignment is done to optimize the DOAE and the LE. We can see how the differences of both approaches are minimal since this only affects to the situations where there are several concurrent events of the same class, which is not very frequent in the STARSS23 dataset.

	Frame-based					Segment-based				
Rank	Submission	$ER_{20°}$	$F_{20°}$	$DOAE_c$	LR_c	Submission	$ER_{20°}$	$F_{20°}$	$DOAE_c$	LR_c
1	Du_NERCSLIP_task3a_1	0.34	59.8%	12.9°	67.5%	Du_NERCSLIP_task3a_1	0.33	62.7%	12.9°	72.1%
2	Liu_CQUPT_task3a_2	0.37	54.0%	13.7°	61.5%	Liu_CQUPT_task3a_2	0.35	58.5%	13.5°	65.7%
3	Yang_IACAS_task3a_2	0.36	50.2%	16.3°	61.0%	Yang_IACAS_task3a_2	0.35	54.5%	15.8°	66.7%
4	Kang_KT_task3a_2	0.41	48.0%	15.3°	60.7%	Kang_KT_task3a_2	0.40	51.4%	15.0°	63.8%
5	Kim_KU_task3a_4	0.46	46.1%	14.9°	58.1%	Kim_KU_task3a_4	0.45	49.0%	15.0°	62.5%

Table 1: Comparison of the frame-based and segment-based metrics applied to the system of the challenge 2023.

Assignation	F_{20°	$DOAE_c$	RDE_c	LE_c [cm]
DOAE	18.0%	29.6°	0.31	137.6
LE	17.9%	29.7°	0.31	137.4

Table 2: 2024 audio-only baseline results when the assignment between estimates and references of concurrent events of the same class are done to minimize the DOAE or the LE.

5. RESULTS

Incorporating all the changes, Table 3 summarizes the results of the baseline models on the STARSS23 evaluation dataset trained for the SELD task along with distance estimation with the new frame-based metrics using the Multi-ACCDDOA loss. The performance of the models on both 4-channel ambisonic (FOA) and tetrahedral microphone array (MIC) audio formats are presented for comparison.

Dataset	Format	F_{20°	$DOAE_c$	RDE_c
Audio	FOA	18.0%	29.6°	0.31
Audio-visual	FOA	15.5%	34.7°	0.31
Audio	MIC-GCC	16.0%	34.2°	0.30
Audio-visual	MIC-GCC	15.8%	36.0°	0.30

Table 3: Baseline results on STARSS23 evaluation dataset.

Compared with the baselines of the previous edition of the challenge, we can observe a reduction in the performance of the audio-only system. This is due to 1. the addition of the distance estimation task, which makes the problem harder, and 2. the changes in the training pipeline, where synthetic data was only used to pre-train the audio feature extraction layers as done in the audio-visual system. On the other hand, the performance of the audio-visual system has clearly improved compared to the previous edition of the challenge thanks to the changes done in the visual feature extraction, so we are narrowing the gap between both systems. However, further research is still needed to really exploit the visual information of the 360° video input.

6. CONCLUSIONS

This report highlights the changes introduced in the SELD task of DCASE2024 challenge. Most of the changes on baseline models, and task evaluation are associated to the newly-introduced distance estimation objective of the challenge. Distance estimation with a single compact array makes the task significantly more challenging as can be observed from the low baseline results for both audio-only and audiovisual tracks. Training losses and metrics are adapted in order to accommodate the new objective effectively. Audiovisual processing for the currently proposed baseline remains inferior to the baseline using only audio input.

References

[1] S. Adavanne, A. Politis, J. Nikunen, and T. Virtanen, “Sound event localization and detection of overlapping sources using convolutional recurrent neural networks,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 1, p. 34–48, Mar. 2019. [Online]. Available: http://dx.doi.org/10.1109/JSTSP.2018.2885636

[2] A. Politis, A. Mesaros, S. Adavanne, T. Heittola, and T. Virtanen, “Overview and evaluation of sound event localization and detection in dcase 2019,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 684–698, 2020. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9306885

[3] A. Politis, S. Adavanne, and T. Virtanen, “A dataset of reverberant spatial sound scenes with moving sources for sound event localization and detection,” in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020)*, Tokyo, Japan, November 2020, pp. 165–169.

[4] K. Shimada, A. Politis, P. Sudarsanam, D. Krause, K. Uchida, S. Adavanne, A. Hakala, Y. Koyama, N. Takahashi, S. Takahashi, *et al.*, “STARSS23: An audio-visual dataset of spatial recordings of real scenes with spatiotemporal annotations of sound events,” in *Proc. of NeurIPS*, 2023.

[5] Y. Cao, T. Iqbal, Q. Kong, Y. Zhong, W. Wang, and M. D. Plumbley, “Event-independent network for polyphonic sound event localization and detection,” in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020)*, Tokyo, Japan, November 2020, pp. 11–15.

[6] K. Shimada, Y. Koyama, N. Takahashi, S. Takahashi, and Y. Mitsufuji, “Accdoa: Activity-coupled cartesian direction of arrival representation for sound event localization and detection,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 915–919.

[7] T. N. T. Nguyen, D. L. Jones, and W.-S. Gan, “A sequence matching network for polyphonic sound event localization and detection,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 71–75.

[8] Q. Wang, J. Du, H.-X. Wu, J. Pan, F. Ma, and C.-H. Lee, “A four-stage data augmentation approach to resnet-conformer based acoustic modeling for sound event localization and detection,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 1251–1264, 2023.

[9] A. Politis, K. Shimada, P. Sudarsanam, S. Adavanne, D. Krause, Y. Koyama, N. Takahashi, S. Takahashi, Y. Mitsufuji, and T. Virtanen, “Starss22: A dataset of spatial recordings of real scenes with spatiotemporal annotations of sound events,” in *Proceedings of the 7th Detection and Classification of Acoustic Scenes and Events 2022 Workshop (DCASE2022)*, Nancy, France, November 2022.

[10] S. Niu, J. Du, Q. Wang, L. Chai, H. Wu, Z. Nian, L. Sun, Y. Fang, J. Pan, and C.-H. Lee, “An experimental study on sound event localization and detection under realistic testing conditions,” in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[11] Y. Shul and J.-W. Choi, “Cst-former: Transformer with channel-spectro-temporal attention for sound event localization and detection,” in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2024, pp. 8686–8690.

[12] M. Yiwere and E. J. Rhee, “Sound source distance estimation using deep learning: An image classification approach,” *Sensors*, vol. 20, no. 1, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/1/172

[13] A. Sobhdel, R. Razavi-Far, and S. Shahrivari, “Few-shot sound source distance estimation using relation networks,” 2021. [Online]. Available: https://arxiv.org/abs/2109.10561

[14] S. S. Kushwaha, I. R. Román, M. Fuentes, and J. P. Bello, “Sound source distance estimation in diverse and dynamic acoustic conditions,” in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, WASPAA 2023.* IEEE, pp. 1–5.

[15] M. Yiwere and E. J. Rhee, “Distance estimation and localization of sound sources in reverberant conditions using deep neural networks,” in *2017 International Journal of Applied Engineering Research*, 2017.

[16] D. A. Krause, A. Politis, and A. Mesaros, “Joint direction and proximity classification of overlapping sound events from binaural audio,” in *2021 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA).* IEEE, 2021, pp. 331–335.

[17] D. A. Krause, G. García-Barrios, A. Politis, and A. Mesaros, “Binaural sound source distance estimation and localization for a moving listener,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 996–1011, 2024.

[18] D. A. Krause, A. Politis, and A. Mesaros, “Sound event detection and localization with distance estimation,” in *32nd European Signal Processing Conference (EUSIPCO).* (accepted - preprint arXiv:2403.11827), 2024.

[19] K. Shimada, Y. Koyama, S. Takahashi, N. Takahashi, E. Tsunoo, and Y. Mitsufuji, “Multi-accdoa: Localizing and detecting overlapping sounds from the same class with auxiliary duplicating permutation invariant training,” in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 316–320.

[20] P. Sudarsanam, A. Politis, and K. Drossos, “Assessment of self-attention on learned features for sound event localization and detection,” in *Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2021.

[21] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, “YOLOX: Exceeding YOLO series in 2021,” *arXiv preprint arXiv:2107.08430*, 2021.

[22] X. Qian, Z. Wang, J. Wang, G. Guan, and H. Li, “Audio-visual cross-attention network for robotic speaker tracking,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 550–562, 2022.

[23] D. Berghi, P. Wu, J. Zhao, W. Wang, and P. J. Jackson, “Fusion of audio and visual embeddings for sound event localization and detection,” in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2024, pp. 8816–8820.

[24] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[26] I. R. Roman, C. Ick, S. Ding, A. S. Roman, B. McFee, and J. P. Bello, “Spatial scaper: a library to simulate and augment soundscapes for sound event localization and detection in realistic rooms,” in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2024, pp. 1221–1225.

[27] A. Mesaros, S. Adavanne, A. Politis, T. Heittola, and T. Virtanen, “Joint measurement of localization and detection of sound events,” in *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2019, pp. 333–337.

[28] A. Mesaros, T. Heittola, and T. Virtanen, “Metrics for polyphonic sound event detection,” *Applied Sciences*, vol. 6, no. 6, 2016. [Online]. Available: https://www.mdpi.com/2076-3417/6/6/162

[29] H. W. Kuhn, “The hungarian method for the assignment problem,” *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

FROM COMPUTATION TO CONSUMPTION: EXPLORING THE COMPUTE-ENERGY LINK FOR TRAINING AND TESTING NEURAL NETWORKS FOR SED SYSTEMS

Constance Douwes, Romain Serizel

University of Lorraine, CNRS, Inria, Loria, 54000, Nancy, France
constance.douwes@inria.fr, romain.serizel@loria.fr

ABSTRACT

The massive use of machine learning models, particularly neural networks, has raised serious concerns about their environmental impact. Indeed, over the last few years we have seen an explosion in the computing costs associated with training and deploying these systems. It is, therefore, crucial to understand their energy requirements in order to better integrate them into the evaluation of models, which has so far focused mainly on performance. In this paper, we study several neural network architectures that are key components of sound event detection systems, using an audio tagging task as an example. We measure the energy consumption for training and testing small to large architectures and establish complex relationships between the energy consumption, the number of floating-point operations, the number of parameters, and the GPU/memory utilization.

***Index Terms*—** Energy, deep learning, neural networks, FLOPs, parameters, training, inference, sound event detection

1. INTRODUCTION

Deep learning (DL) has become the principal focus of audio processing research, with numerous applications spanning various domains including sound event detection (SED) [1, 2], speech recognition [3, 4] and music generation [5, 6]. As models become increasingly powerful and datasets grow larger, the associated computational costs have exploded [7, 8, 9]. Yet, the true cost of computation often remains obscured, as many computations are carried out on remote infrastructures or data centers. Nevertheless, these energy-intensive processes involved in training and deploying high-performance models have a real environmental footprint linked to their demand for electricity [10, 11]. This raises significant concerns in the current context of climate change and efforts to limit global warming to below 2 degrees [12]. Even though models used in audio processing are smaller than those used in natural language processing, they still present similar problems [13, 14].

The trends described above are driven by an ongoing pursuit of outperforming previous state-of-the-art systems, even by a small margin. Recently, there has been a slight shift towards reporting and quantifying the environmental costs associated with these advances [15, 16]. In the audio processing domain in particular, significant efforts have been made to balance performance and energy in the context of sound event detection [17, 18] or speech recognition [14], and to emphasize the importance of considering quality metrics alongside energy footprint assessments in speech synthesis [13]. All of these studies call for a fair and reliable metric to assess the computational footprint that reflects the energy consumption while being hardware independent to enable accurate comparisons between models. Although work such as Speckhard et al. [19] shows a strong correlation between computational cost and energy consumption during inference for convolution-based models, to our knowledge similar investigations have not been conducted for training or for other architectures. Even if a few hundred experiments are sometimes required to train a model, the cost of the training phase represents only 10% to 20% of the total CO2 emissions of the associated machine learning usage, with the majority occurring during the inference phase [20]. However, as audio processing researchers, the majority of our energy consumption lies in the training phase, and should not be overshadowed.

In this article, we aim to understand the computational factors that impact the energy consumption for the training or testing deep learning models that compose SED systems. This study is conducted in the context of the DCASE challenge task 4, where participants have been required since 2022 [21] to report their energy consumption alongside computational factors such as the number of parameters and the number of operations. Specifically, we seek an indicator that can estimate the energy consumption based on computational measurements. This would allow us to estimate each system's consumption on the same hardware and provide fair comparisons between systems, extending the work of Ronchini et al.[18]. We focus our analysis on well-known architectures such as MLP, RNN, CNN and CRNN. CRNN is specifically the current architecture used in Task 4 of the DCASE Challenge [22]. We compute the number of parameters of the models and the number of floating point operations (FLOPs) as two potential candidate factors for energy consumption estimation. We show that as the number of operations increases, so does the energy consumption across all architectures during both the test and training phases. However, the relative increase in energy consumption varies between architectures and phases. We identify two distinct trends: one for MLP/RNN, and one for CNN/CRNN. Finally, we identify relationship between energy consumption and GPU utilization during both training and testing phases, which could serve as a basis for future research on computational metrics.

In summary, our key contributions are :

- A comparative analysis of prominent architectures (MLP, CNN, RNN, CRNN) and their associated energy consumption.
- The identification of two distinct trends in energy consumption based on architecture type, notably distinguishing between MLP/RNN and CNN/CRNN architectures.
- A relative comparison of power usage between training and test stages.

2. METHODOLOGY

Computing and monitoring the computational and energy costs of the two phases of deep learning systems - training and inference

- is a complex endeavour. We present here our methodology for assessing both, mentioning previous work in these areas.

2.1. Computational cost

Traditional methods rely on metrics such as the size of the model (the number of parameters) and the number of floating-point operations (FLOPs) computed by the model to estimate the computational cost. While computing the number of parameters (or weights) of a model is straightforward, computing the number of operations can be a difficult task, especially for complex architectures, and this number is very sensitive to the size of the input/output. At inference, only forward calculations are performed, so the number of operations is the sum of all operations across all layers. We use the deepspeed profiler [23] to quantify these forward pass operations accurately. In contrast, training is a more complex process involving iterative forward and backward calculations. In particular, the backward pass also computes the gradient with respect to the parameters, the loss and update the weights. However, at the time of writing, no profiler provided the exact number of backward operations, so we derive this number using the ratio 2:1 as an approximation [24]. In total, the number of operations of a training iteration (forward and backward) is three times the number of operations of an inference (forward only).

2.2. Energy consumption

Several Python trackers have emerged to facilitate the computation of energy consumption [25]. In most of the trackers, the total consumption is calculated as the sum of the consumption of each component of the computer: GPU, CPU and RAM. In our study, we focus specifically on analysing the energy consumption of the GPU given by CodeCarbon [26]. Indeed, preliminary experiments have led us to conclude that while GPU power fluctuates, CPU power remains stable. Regarding ram energy, CodeCarbon estimates 3 watts per 8 GB, which also remains constant over time. We made sure that any increases in GPU power with the python trackers were correlated with energy consumption monitored on the system's baseboard management controller (BMC). We also monitor the GPU and memory utilization from Nvidia SMI query every 5 seconds to get the mean uses of the each experiment.

3. EXPERIMENTS

Our objective is to better understand the energy consumption at train and test and to relate it to computational cost of a given model and architecture. To achieve this, we evaluate different types and sizes of architectures for audio tagging systems.[1]

3.1. Task description

Audio tagging involves assigning one or multiple tags to an audio signal without any temporal information. For this experiment, we work on the real part of the DESED dataset [27]. This dataset contains 10-second audio clips recorded in domestic environments. We convert those recordings into mel-spectrogram representations with 128 bands, an FFT size of 2048 and a hop size of 256. We only take the first 64 frames as input, which corresponds to approximately the first 1 second of the audio signal. Although this significantly impacts the performance of the model, it reduce the system's complexity, allowing for more lightweight experiments, as we do not focus on performance but only on energy.

Model	Num Layers	Hidden Sizes
MLP	1	512, 1024, 2048
	4	1024, 2048, 4096
	6, 10, 16, 32	4096
CNN	1	128, 256, 512, 1024
	2	128, 256, 384, 512, 768, 1024
	6	384, 768
RNN	1	128, 512, 1024, 2048
	4, 6	1024, 2048
	2, 10, 14	2048
CRNN	[1,1], [2,1], [1,2]	[64,64], [256,64], [512, 256]
	[2,2]	[728, 256]
	[1,2], [2,2]	[1024, 256]

Table 1: Summary of all the configurations tested in our experiment. For each number of layer, we tested different hidden sizes. For CRNN, the configurations first indicate the convolutional layers and then the recurrent layers.

3.2. Models

We implement four neural network architectures: multi-layer perceptron (MLP), convolutional neural network (CNN), recurrent neural network (RNN), and convolutional recurrent neural network (CRNN). For the MLP, we implement a series of linear layers followed by ReLU activation functions. For the CNN, we adopt a sequence of Conv2d, ReLU and MaxPool2d layers. For the RNN we use GRU cells and for the CRNN we start with Conv2d, ReLU and MaxPool2d layers followed by a GRU cell. All implementations are completed with a final linear layer and a sigmoid activation function that outputs a probability vector for the 10 classes. For each architecture, we systematically increase the number of layers and adjust the hidden sizes per layer, gradually scaling up to reach the full GPU memory capacity and utilization, resulting in 43 models. We present the summary of all the configurations tested in Table 1. We intentionally chose those configurations to achieve meaningful variations in the number of FLOPs without conducting redundant experiments.

3.3. Training and test

Our experiments diverge from the conventional research of accuracy performance. Instead, we train all models for a single epoch on the same Nvidia Tesla T4 GPU and monitor the energy of the training phase. To focus solely on architectural differences, we use a consistent batch size of 8. Although the choice of criterion, optimizer, and learning rate is crucial for model convergence, it does significantly impact energy measurements. Therefore, we employ the cross-entropy function as the criterion, fix the learning rate at 10^{-3}, and use the ADAM optimizer [28]. We did not include any validation steps in the training routine to isolate the effects of training. Instead, we measure the energy consumption during the test phase separately. The test phase involves running the model (inference) and computing the error. Although inference for such small models can generally be performed on the CPU, we ensure consistency with the training phase measurements by also running the

[1] `https://github.com/ConstanceDws/toolbox_energy`

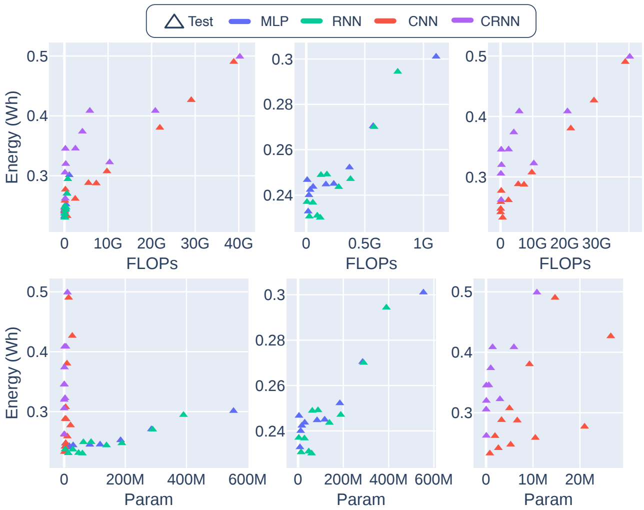


Figure 1: Energy consumption at test for various neural network architectures and configurations, as a function of FLOPs (top) and of parameters (bottom). The three columns show: (1) all architectures together, (2) only MLP/RNN (in blue and green), and (3) only CNN/CRNN (in red and purple).

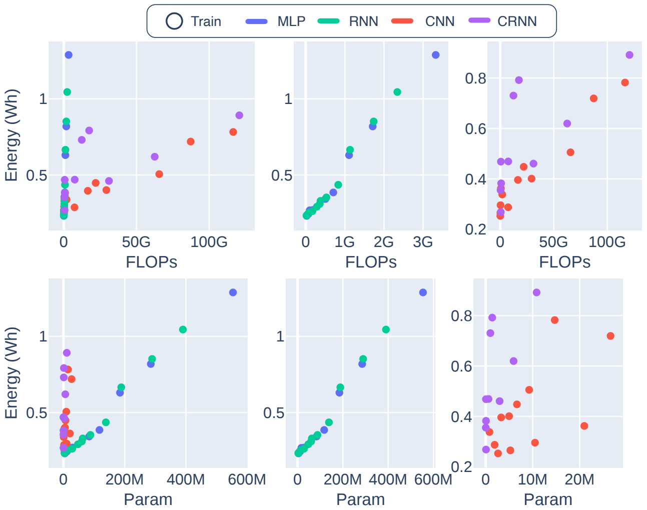


Figure 2: Energy consumption for training various neural network architectures and configurations, as a function of FLOPs (top) and of parameters (bottom). The three columns show: (1) all architectures together, (2) only MLP/RNN (in blue and green), and (3) only CNN/CRNN (in red and purple).

test phase on the same Nvidia T4 GPU for the entire dataset (corresponding to 1 epochs of training).

4. RESULTS

In this section, we explore the relationship between computational metrics and the energy consumption. Our analysis aims to identify trends and discrepancies in energy consumption at train and test across various architectures and configurations.

4.1. Relationship between energy and computational cost at test

First, we examine the energy consumption of the test, as existing research suggests that there is a correlation between FLOPs and energy consumption for convolutional models [19] on CPU. Figure 1 shows the result of this experiment, where the top row presents the GPU energy consumption as a function of FLOPs, and the bottom row the energy consumption as a function of the number of parameters. The first row shows that increasing the number of operations at test leads to an increase in energy consumption for all types of architecture. A closer examination of each architecture type reveals that the relationship between FLOPs and energy consumption exhibits some affine patterns. Examining the number of parameters in the second row, significant disparities emerge between MLP/RNN and CNN/CRNN models: the relationship between the number of parameters and the energy consumption is almost affine for MLP/RNN (and similar to the relationship with FLOPs), but for CNN and CRNN the relationship is more chaotic. This discrepancy is mainly due to the architectural elements composing these networks. Convolutional layers use parameter sharing, which contrasts with fully connected layers where each parameter is unique to its connection. Similarly, in recurrent layers, the connections between units often have unique weights, although some forms of parameter sharing can occur as well. Consequently, MLP and RNN have a higher number of parameters but a lower number of operations relative to CNN. These observations suggest that the number of operations and the number of parameters are not reliable indicators for estimating energy consumption at test, regardless of the model type, as the affine patterns are not consistent across architectures. However, they could be useful within a single architecture scenario comparisons.

4.2. Relationship between energy and computational cost at training

Building on our previous results, we now investigate the energy consumption associated with training. Figure 2 displays the energy consumption for training in function of the two computational metrics arranged as previously described. Regarding the interaction between energy and FLOPs, we observe two distinct trends. For MLP/RNN, the data points follow a steep curve on the left side, while for CNN, the curve smoothly increases and spans the entire plot. The CRNN architecture appears to exhibit characteristics that lie between the two aforementioned trends. In some configurations, the CRNN behaves as a CNN at higher FLOPs and as an RNN at lower FLOPs. A plausible explanation of this two trends could be the higher memory exchanges associated with MLP/RNN compared to CNN architectures that would cause higher energy consumption but do not increase the FLOPs. An important result is the almost affine relationship between FLOPs and energy consumption for MLP and RNN, suggesting that GPUs handle these architectures similarly during training causing close energy consumption for the same FLOPs. However, for CNN and CRNN, FLOPs alone do not provide a conclusive estimate of the energy consumption. Regarding the number of parameters, we conclude consistent results as for the test relationship. As a result, for the training consumption, neither FLOPs nor parameters are good estimators of energy consumption without specific knowledge of the model architecture, and one hypothesis could comes from the difference between the archi-

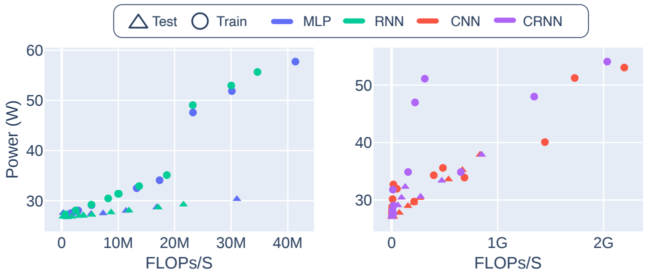


Figure 3: Average power during training (circles) and test (triangles) as a function of FLOPs/S.

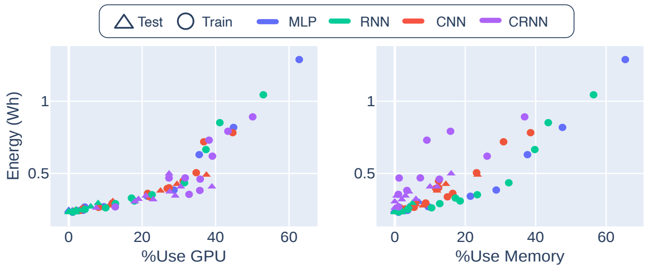


Figure 4: Relationship between the energy consumption and the GPU utilization (left) and memory utilization (right) for training and test.

tectural elements of the network.

4.3. Training and test comparisons

To further investigate the link between energy and computation, we investigate the mean average power at test and train and relate it to the number of floating points operations per seconds. The average is calculated as the energy divided by the length of the experiment. We present the result of this analysis in Figure 3, where the FLOPs/S is computed as the FLOPs divided by the duration of one epoch for training and test. We see that there is a nearly-affine relationship between FLOPs/S and power at test for the MLP/RNN architectures, as indicated by the aligned triangles. However, this affine relationship is less evident for training, as highlighted by a bend around 20M FLOPs/S. An significant result of this analysis is the disparity in average power consumption between MLP/RNN at train and test: circles are positioned higher on the plot, while triangles are lower and there is no overlap between the two sets. In contrast, for CNN and CRNN, triangles and circles occupy similar regions, indicating that MLP and RNN architectures require much more power for training than for testing compared to CNN/CRNN.

4.4. GPU and memory utilization

During our experiments, we also monitored the GPU and memory utilization given by Nvidia SMI. Figure 4 illustrates the relationship between the energy and the GPU and memory utilization during both training and test phases. Notably, a strong correlation exists between GPU use and energy. What is noteworthy is that this correlation remains independent of the phase (train or test) and the architectures. This results in a metric that is highly recommended for estimating the energy consumption of a given model, although it is dependent on the hardware. It would be interesting to find a combination of the number FLOPs and the number of parameters that could reflect the GPU utilization. For memory utilization, the correlation is not as straightforward, but it shows that memory also has an impact on energy consumption, with a higher dependency on the architecture type than GPU utilization.

5. DISCUSSION AND FUTURE WORKS

In this article, we specifically study the audio tagging task, using very simple architectures that are far from current SED models. It would therefore be interesting to explore more advanced models in the field and assess whether similar trends persist. In addition, the training procedure implemented here is one of the most conventional methods of deep learning, but recent advances have introduced much more complex procedures, resulting in higher computational costs and potentially different energy consumption. For example, using techniques such as teacher-student learning (used in the baseline) can lead to higher computational costs and therefore a different energy footprint. It is also important to note that energy consumption throughout our study is measured for a single epoch, and is therefore relative to the dataset. Experiments to determine whether there is a linear relation between data size and energy consumption would be recommended to remove the dependency on the dataset.

Additionally, we focused here on a single hardware (one Nvidia Tesla T4). However, analyzing the differences within a single hardware configuration and exploring the variations between different hardware configurations could provide some additional information on the energy consumption. This approach could also contribute to efforts to normalize hardware energy measurements, such as those proposed by Serizel et al. [17]. Furthermore, our study did not address the performance of the models. It's likely that a CNN and CRNN may have different performances compared to an MLP or an RNN. This concept aligns with Douwes et al. [29], emphasizing the need to explore multi-objective criteria by considering factors such as model performance, energy consumption, and computational efficiency simultaneously.

6. CONCLUSIONS

Our study provides a better understanding of the relationship between computational cost and energy consumption for various neural networks used in SED tasks. We observed that while the number of floating-point operations and the number of parameters influenced energy consumption, these metrics were not consistent predictors across all architectures. We identify distinct trends and discrepancies in energy consumption during both testing and training phases, with notable differences between MLP/RNN and CNN/CRNN models. Finally, we establish correlations between energy consumption and GPU utilization for both training and test phases, that could lay as a foundation for future research on computational indicators. We hope that this study will contribute to the development of green AI practices not only in speech processing but also across other domains.

7. REFERENCES

[1] J. W. Kim, S. W. Son, Y. Song, H. K. Kim, I. H. Song, and J. E. Lim, “Semi-supervsied learning-based sound event de-

tection using freuqency dynamic convolution with large kernel attention for dcase challenge 2023 task 4," *arXiv preprint arXiv:2306.06461*, 2023.

[2] D. Berghi, P. Wu, J. Zhao, W. Wang, and P. J. Jackson, "Fusion of audio and visual embeddings for sound event localization and detection," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 8816–8820.

[3] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12 449–12 460, 2020.

[4] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *International Conference on Machine Learning*. PMLR, 2023, pp. 28 492–28 518.

[5] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi, *et al.*, "Musiclm: Generating music from text," *arXiv preprint arXiv:2301.11325*, 2023.

[6] A. Caillon and P. Esling, "Rave: A variational autoencoder for fast and high-quality neural audio synthesis," *arXiv preprint arXiv:2111.05011*, 2021.

[7] D. Amodei, D. Hernandez, G. Sastry, J. Clark, G. Brockman, and I. Sutskever, "Ai and compute," 2018.

[8] J. Sevilla, L. Heim, A. Ho, T. Besiroglu, M. Hobbhahn, and P. Villalobos, "Compute trends across three eras of machine learning," in *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2022, pp. 1–8.

[9] N. C. Thompson, K. Greenewald, K. Lee, and G. F. Manso, "The computational limits of deep learning (2020)," *arXiv preprint arXiv:2007.05558*, 2007.

[10] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for deep learning in nlp," *arXiv preprint arXiv:1906.02243*, 2019.

[11] A. S. Luccioni, S. Viguier, and A.-L. Ligozat, "Estimating the carbon footprint of bloom, a 176b parameter language model," *Journal of Machine Learning Research*, vol. 24, no. 253, pp. 1–15, 2023.

[12] United Nations, "Paris Agreement," 2015.

[13] C. Douwes, G. Bindi, A. Caillon, P. Esling, and J.-P. Briot, "Is quality enough: Integrating energy consumption in a large-scale evaluation of neural audio synthesis models," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[14] T. Parcollet and M. Ravanelli, "The energy and carbon footprint of training end-to-end speech recognizers," 2021.

[15] P. Henderson, J. Hu, J. Romoff, E. Brunskill, D. Jurafsky, and J. Pineau, "Towards the systematic reporting of the energy and carbon footprints of machine learning," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 10 039–10 081, 2020.

[16] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, "Green ai," *Communications of the ACM*, vol. 63, no. 12, pp. 54–63, 2020.

[17] R. Serizel, S. Cornell, and N. Turpault, "Performance above all? energy consumption vs. performance, a study on sound event detection with heterogeneous data," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[18] F. Ronchini and R. Serizel, "Performance and energy balance: a comprehensive study of state-of-the-art sound event detection systems," *arXiv preprint arXiv:2310.03455*, 2023.

[19] D. T. Speckhard, K. Misiunas, S. Perel, T. Zhu, S. Carlile, and M. Slaney, "Neural architecture search for energy efficient always-on audio models," *arXiv preprint arXiv:2202.05397*, 2022.

[20] C.-J. Wu, R. Raghavendra, U. Gupta, B. Acun, N. Ardalani, K. Maeng, G. Chang, F. Aga, J. Huang, C. Bai, *et al.*, "Sustainable ai: Environmental implications, challenges and opportunities," *Proceedings of Machine Learning and Systems*, vol. 4, pp. 795–813, 2022.

[21] F. Ronchini, S. Cornell, R. Serizel, N. Turpault, E. Fonseca, and D. P. Ellis, "Description and analysis of novelties introduced in dcase task 4 2022 on the baseline system," *arXiv preprint arXiv:2210.07856*, 2022.

[22] F. Ronchini, J. Ebbers, F. Angulo, D. Perera, S. Essid, and R. Serizel, "DCASE 2023 Task 4a Challenge," https://dcase.community/challenge2023, 2023.

[23] J. Rasley, S. Rajbhandari, O. Ruwase, and Y. He, "Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 3505–3506.

[24] M. Hobbhahn and J. Sevilla, "What's the backward-forward flop ratio for neural networks?" 2021, accessed: 2024-03-01. [Online]. Available: https://epochai.org/blog/backward-forward-FLOP-ratio

[25] M. Jay, V. Ostapenco, L. Lefèvre, D. Trystram, A.-C. Orgerie, and B. Fichel, "An experimental comparison of software-based power meters: focus on cpu and gpu," in *CCGrid 2023-23rd IEEE/ACM international symposium on cluster, cloud and internet computing*. IEEE, 2023, pp. 1–13.

[26] V. Schmidt, K. Goyal, A. Joshi, B. Feld, L. Conell, N. Laskaris, D. Blank, J. Wilson, S. Friedler, and S. Luccioni, "Codecarbon: estimate and track carbon emissions from machine learning computing (2021)," *DOI: https://doi.org/10.5281/zenodo*, vol. 4658424, 2021.

[27] N. Turpault, R. Serizel, A. P. Shah, and J. Salamon, "Sound event detection in domestic environments with weakly labeled data and soundscape synthesis," in *Workshop on Detection and Classification of Acoustic Scenes and Events*, 2019.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[29] C. Douwes, P. Esling, and J.-P. Briot, "Energy consumption of deep generative audio models," *arXiv preprint arXiv:2107.02621*, 2021.

IMPROVING QUERY-BY-VOCAL IMITATION WITH CONTRASTIVE LEARNING AND AUDIO PRETRAINING

Jonathan Greif[1], *Florian Schmid*[1], *Paul Primus*[1], *Gerhard Widmer*[1,2]

[1]Institute of Computational Perception (CP-JKU), [2]LIT Artificial Intelligence Lab,
Johannes Kepler University Linz, Austria
{jonathan.greif, florian.schmid, paul.primus}@jku.at

ABSTRACT

Query-by-Vocal Imitation (QBV) is about searching audio files within databases using vocal imitations created by the user's voice. Since most humans can effectively communicate sound concepts through voice, QBV offers the more intuitive and convenient approach compared to text-based search. To fully leverage QBV, developing robust audio feature representations for both the vocal imitation and the original sound is crucial. In this paper, we present a new system for QBV that utilizes the feature extraction capabilities of Convolutional Neural Networks pre-trained with large-scale general-purpose audio datasets. We integrate these pre-trained models into a dual encoder architecture and fine-tune them end-to-end using contrastive learning. A distinctive aspect of our proposed method is the fine-tuning strategy of pre-trained models using an adapted NT-Xent loss for contrastive learning, creating a shared embedding space for reference recordings and vocal imitations. The proposed system significantly enhances audio retrieval performance, establishing a new state of the art on both coarse- and fine-grained QBV tasks[1].

Index Terms— audio retrieval, vocal imitation, dual encoder, contrastive learning, QBV

1. INTRODUCTION

Traditional audio retrieval systems rely on textual descriptions or keywords to search for audio recordings (e.g., [1, 2, 3, 4]). Those descriptors are well suited to describe acoustic events on a high level. However, conveying specific acoustic nuances, such as pitch, loudness, timbre, or temporal relationships, via textual descriptions is difficult. For example, non-experts might struggle to find the right vocabulary to describe specific computer-synthesized sound effects. However, since most humans can effectively imitate acoustic events through their vocal tract, Query-by-Vocal Imitation (QBV) becomes an attractive alternative. In fact, previous work has suggested that QBV-based search engines actually achieve higher user satisfaction than text-based search engines [5].

Previous work on QBV systems such as TL-IMINET [6] and M-VGGish [7] relied on custom or relatively outdated pretrained audio embedding models and simple (non-contrastive) loss functions for training. Pishdadian et al. [8] showed that simple signal processing methods based on handcrafted features outperformed those systems in their experimental setups [8]. In this work, we leverage contrastive training and the feature extraction capabilities of a more recent, pre-trained Convolutional Neural Network (CNN) model in a dual encoder architecture to improve QBV. The approach is sketched in Figure 1. Experiments conducted on VimSketch [8] and VocalImitationSet [9] demonstrate that our method outperforms the previous deep-learning-based approaches and the handcrafted approach (Sections 4.1 and 4.2). We further conducted an ablation study in Section 5.3 to measure the impact of each of our proposed method's design choices.

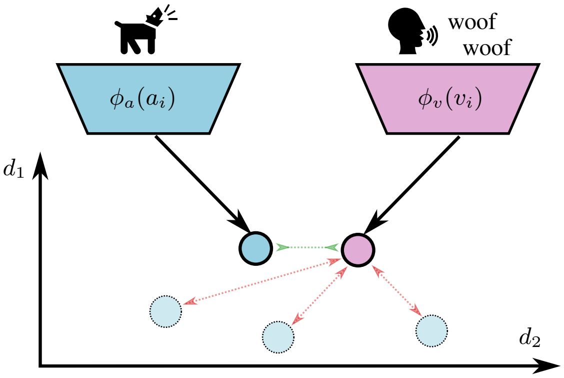


Figure 1: Two separate audio embedding models ϕ_a and ϕ_v project the reference sounds a and the vocal imitations v into a shared metric space. The contrastive loss increases the similarity of vocal imitations and their corresponding sounds while pushing mismatching pairs away from each other in this metric space.

2. RELATED WORK

Query-by-Vocal Imitation (QBV) is a special case of Query-by-Example (QBE) [10]. QBE encompasses various audio retrieval tasks such as cover song recognition [11], Query-by-Beatboxing [12], and Query-by-Humming [13, 14]. Unlike these music-related applications, QBV specifically focuses on general sound search.

Among the most recent advancements in QBV are systems like TL-IMINET [6] and CR-IMINET [5]. Those are based on CNN-based dual encoder architectures, which rely on two separate embedding towers for the two domains (real and imitated sounds). Instead of comparing the embedding vectors directly, [5, 6] incorporate a Feedforward Neural Network (FNN) that takes the embedding vectors as input and outputs an estimate of their similarity. TL-IMINET distinguishes itself by employing transfer learning, while CR-IMINET incorporates a Recurrent Neural Network layer. Another noteworthy system used for QBV is M-VGGish [7], which combines features extracted from intermediate layers of VGGish [15]. The model was pre-trained for audio tagging but not fine-tuned with imitations and reference sounds for

[1] Code at: `https://github.com/Jonathan-Greif/QBV`

QBV. M-VGGish assesses similarities by measuring the cosine similarity between the embedding vectors. On the VocalSketch [16] dataset M-VGGish demonstrated superior performance compared to TL-IMINET, highlighting the feature extraction capabilities of models pre-trained on large audio tagging datasets [7]. However, recent work showed that these latest QBV systems perform poorly compared to simple signal processing (SP) methods in certain settings [8]. The most performant of these SP methods involved converting the signals into the frequency domain using the Constant-Q Transform (CQT) and further with a 2D Fourier transformation (2DFT). The resulting representations were then compared using the cosine similarity.

3. PROPOSED SYSTEM

Similarly to previous methods [6, 5, 7], our system relies on two separate audio embedding models to project reference sounds a_i and vocal imitations v_i thereof, into a shared embedding space (see Figure 1). In the following, we will denote the model that is used to embed the reference and the imitated acoustic events as ϕ_a and ϕ_v, respectively. The correspondence between a vocal imitation v_i and a reference sounds a_i is determined by their distance in the embedding space. Our proposed system improves over previous deep-learning-based QBV solutions in two main aspects, namely, the audio embedding model and the fine-tuning strategy on reference sound and imitation pairs using contrastive learning. In Section 3.1 we motivate the choice for the audio embedding model and in Section 3.2 we describe how these models are fine-tuned using contrastive learning to align vocal imitations and reference sounds in the shared embedding space.

3.1. Audio Embedding Model

Extracting high-quality audio embeddings is a fundamental building block of a well-performing QBV system. The quality of these embeddings extracted by deep learning systems depends both, on the neural network architecture and the audio dataset it has been trained on. Previous QBV systems used small, custom architectures (e.g., TL-IMINET [6] or CR-IMINET [5]), or architectures that are outdated from today's point of view (M-VGGish [7]). These architectures were either directly trained on vocal imitation-reference sound pairs [5], pre-trained on smaller domain-specific datasets, and then fine-tuned on vocal imitation-reference sound pairs [6], or pre-trained on larger audio tagging datasets [7] but not fine-tuned on vocal imitation-reference sound pairs.

Our approach uses MobileNetV3 (MN) [17], a modern efficient CNN pre-trained on AudioSet [18], as an audio embedding model. AudioSet is a large general-purpose audio dataset, consisting of 2 million 10-second audio clips labeled with 527 sound event classes. MNs achieve highly competitive performance on AudioSet when trained with Knowledge Distillation [19] from a large transformer ensemble [20]. Additionally, pre-trained MNs have been shown to extract high-quality audio embeddings across music, environmental sound, and speech tasks [21]. We hypothesize that the general audio feature extraction capabilities obtained from AudioSet pre-training renders MN a strong choice for both the reference sound (ϕ_a) and the vocal imitation (ϕ_v) tower in our dual encoder setup.

3.2. Contrastive Learning

Typical training datasets for QBV consist of N pairs, each holding a recording of a reference sound and its corresponding vocal imitation, i.e., $\{(a_i,\ v_i)\}_{i=1}^N$. During training, the two audio embedding networks learn to map inputs into a shared D-dimensionl space in which vocal imitations live close to their corresponding reference sounds. This alignment is achieved through contrastive training, which brings the embeddings of matching pairs $(a_i,\ v_i)$ together while pushing the representations of non-matching pairs $(a_i,\ v_{j;j\neq i})$ apart. The correspondence between a vocal imitation v_i and reference sound a_i is determined using the cosine similarity between the embedded vectors in the shared embedding space:

$$S_{ij} = \frac{\phi_{\text{a}}\left(a_i\right)^T \cdot \phi_{\text{v}}\left(v_j\right)}{\left\|\phi_a\left(a_i\right)\right\|^2 \left\|\phi_v\left(v_j\right)\right\|^2}. \tag{1}$$

If each imitation corresponds to exactly one reference sound and vice versa, then the similarity matrix $S \in \mathbb{R}^{N\times N}$ contains the agreement scores for matching pairs along its diagonal, while the off-diagonal elements represent the agreement scores for mismatching pairs. A popular loss function for contrastive training that has not been explored in the QBV context yet is the NT-Xent [22] loss. This loss first converts these similarities into a probability distribution over reference sounds via a temperature-scaled softmax activation. It then minimizes the cross entropy between the estimated distribution and a target distribution. In our case, the target distribution puts the entire probability mass on the reference recording for a given vocal imitation. The corresponding loss is then defined as follows:

$$\mathcal{L} = -\frac{1}{N}\sum_{j=1}^{N} \log \frac{\exp\left(S_{jj}/\tau\right)}{\sum_{i=1}^{N} \exp\left(S_{ij}/\tau\right) \mathbb{1}_{i\neq j}}, \tag{2}$$

where τ is a temperature hyper-parameter.

4. EXPERIMENTAL SETUP

We investigated our system's ability to retrieve the correct audio recording on two levels of granularity: coarse-grained and fine-grained. The corresponding experimental setups are explained below. This section further details the audio embedding model, the training procedure, and the evaluation metrics.

4.1. Coarse-grained QBV

In the course-grained setup, we evaluate the system's ability to recognize acoustic events (e.g., ”dog barking,” ”paper tearing,” or ”thunderstorm”) and correctly connect them across the two domains. The retrieved audios only need to contain the same event as the imitation in this setup to count as a match; specific acoustic properties like pitch, loudness, timing, etc. are not required to match.

We relied on the same experimental setup as in [5] to make our results comparable to theirs. To this end, we trained and evaluated our method on the VimSketch [8] dataset. This dataset contains 542 reference sounds and between 13 and 37 corresponding vocal imitations for each of them. As described in [5], we only used 528 reference sounds and their corresponding imitations and split them into 10 folds, each containing around 52 sound events, for cross-validation. Since the reference sounds mostly belong to distinct categories (i.e., two reference sounds typically don't share the same acoustic event), this setup is well-suited to measure the system's coarse-grained retrieval abilities.

4.2. Fine-grained QBV

In the fine-grained setup, we evaluate the system's ability to retrieve a specific audio recording from a set of candidates that all contain the same acoustic event, e.g., its ability to select the best matching dog bark from a diverse collection of dog barks.

We relied on the same experimental setup as Kim et al. [9] to compare our method to theirs, i.e., we trained the proposed system on VocalSketch v1.0.4 [16] and evaluated it on VocalImitationSet [9]. VocalSketch v1.0.4 includes 240 unique reference sounds and around 18 corresponding vocal imitations for each of them; we used half of the data set for training and the other half for validation. Since the exact training-validation split used in [9] has not been made public, we randomly split the data according to their criteria. VocalImitationSet includes 302 unique reference sounds and around 18 vocal imitations for each. Those imitations were created to match their corresponding reference sounds exactly. In addition, each reference sound is also associated with approximately nine hard negative examples that contain the same acoustic event but differ with respect to other acoustic qualities. We relied on these hard negative examples to asses the systems' abilities to find exact matches among the multiple similar candidate recordings.

4.3. Embedding Networks

As discussed in Section 3.1, we chose efficient MobileNetV3 [17], pre-trained on AudioSet [18], as our embedding model. Specifically, we use a publicly available checkpoint referred to as `mn10_as` (available via GitHub[2]) because it strikes a good balance between computational efficiency and performance on the AudioSet benchmark. For audio pre-processing, we match the original feature extraction pipeline of the pre-trained MN [20] for both the reference sounds and the vocal imitations. Furthermore, we truncated or zero-padded all files to a duration of 10 seconds, aligned with MN's AudioSet pre-training setup.

4.4. Training & Augmentations

We used the Adam [23] as an optimizer featuring a learning rate schedule that includes an exponential warm-up (4 epochs), a constant phase (4 epochs), a linear decrease (14 epochs), followed by a fine-tuning phase (8 epochs). We trained for 30 epochs in total with a batch size of 16. The learning rate was set to 5e-4 and 7e-5 in the coarse-grained and fine-grained training setups, respectively. For the NT-Xent loss, we chose a temperature value of $\tau = 0.07$.

To prevent overfitting, we applied multiple data augmentations on vocal imitations and reference sounds during training. We relied on the following methods:

- Time shifting: We randomly shift the waveform forward or backward within a range of 4000 steps.
- Time masking: The mel-spectrogram representations were randomly time-masked with a maximum length of 400 steps.
- Frequency masking: The mel-spectrogram representations were randomly frequency masked with a maximum of 4 bins.
- Freq-MixStyle [24]: Frequency bands in spectrograms were normalized and denormalized again with mixed frequency statistics of other spectrograms from the same batch. With a probability of 0.3, Freq-MixStyle is applied to a batch and the mixing coefficient was drawn from a Beta distribution $B(0.4, 0.4)$.

4.5. Metric

Aligned with [6, 9, 7, 8, 5], we assessed the retrieval performance with Mean Reciprocal Rank (MRR) [25]:

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_{\ i}}, \tag{3}$$

where Q represents a set of vocal imitation queries and rank$_i$ denotes the rank of the target sound among all sounds for the i-th query. MRR values range from 0 to 1, with higher values indicating better retrieval performance. In addition to the MRR and aligned with [9] we report the Mean Recall@k for k=1 and k=2 (MR@1 & MR@2). This metric reflects the proportion of queries that successfully retrieved the target sound within the top k items in the search results.

5. RESULTS & DISCUSSION

This section presents the retrieval performance of our proposed system for coarse- and fine-grained QBV and a comparison to selected systems from the related work. We additionally conducted an ablation study to understand the impact of our design choices better.

5.1. Coarse-grained QBV

We compared results on the coarse-grained benchmark for M-VGGish, 2DFT, and MN to the results for CR-IMINET and TL-IMINET reported in [5]. Table 2 shows the results. Our method achieved the highest MRR of 0.631, substantially outperforming both hand-crafted approaches like 2DFT [8] (0.308) and previous deep-learning-based methods like CR-IMINET [5] (0.348), TL-IMINET [6] (0.325) and M-VGGish [7] (0.228).

We note that TL-IMINET performed much better than M-VGGish and 2DFT, contrary to previously reported results [7, 8]. This is likely due to the larger training set (476 vs. 120 reference sounds), which benefits models that are trained on imitation and reference pairs (i.e., TL-IMINET) but not those that are not (i.e., M-VGGish, 2DFT).

5.2. Fine-grained QBV

The performance on the fine-grained benchmark is shown in Table 3. We also experimented with training the AudioSet pre-trained MN further with vocal imitations in the training dataset by predicting their corresponding sound classes, as an additional training phase before the contrastive learning stage. When doing so, our proposed system outperformed previous methods and achieved the highest MRR of 0.513, surpassing TL-IMINET (0.356), M-VGGish (0.416), and 2DFT (0.489). Nevertheless, the margin of the signal processing method is smaller compared to coarse-grained QBV. When omitting the supervised pre-training, the performance of MN (0.476) falls behind that of 2DFT. This indicates that the granularity of the embedding space should be further improved to allow better discrimination between recordings that contain the same acoustic event. In the given setup, the methods are not explicitly trained to distinguish fine-grained details in recordings that belong to the same concept. Therefore, we hypothesize that optimizing for such a scenario will result in further performance gains.

5.3. Ablation Study

Our ablation study, detailed in Table 1, demonstrates the effectiveness of our proposed method's components in the coarse-grained setting.

By comparing the MN embeddings of reference sounds and imitations with cosine similarity (without additional training on reference–imitation datasets), our system achieved 0.295 MRR (row 1), which is similar to 2DFT (see Table 2).

This setting also allows a comparison between the pre-trained audio embedding models; when the MN is replaced with VGGish, the MRR dropped from 0.295 to 0.228 (compare row 1 in Table 1

[2] `https://github.com/fschmid56/EfficientAT`

		Supervised Pre-Training		**Contrastive Fine-Tuning**		**Performance**		
Model	Dual	AudioSet	VimSketch	Loss	Similarity	MRR	MR@1	MR@2
MN	✓	✓	-	-	cos	0.295	0.175	0.258
MN	✓	✓	-	BCE	FNN	0.354	0.183	0.306
MN	✓	✓	-	BCE	cos	0.439	0.26	0.43
MN	✓	✓	✓	-	cos	0.508	0.35	0.497
MN	✓	✓	✓	BCE	cos	0.582	0.422	0.595
MN	✓	✓	✓	NT-Xent	cos	0.614	0.463	0.619
MN	✓	✓	-	NT-Xent	cos	**0.635**	**0.478**	**0.649**
MN	✓	-	-	NT-Xent	cos	0.493	0.322	0.477
MN	-	✓	✓	NT-Xent	cos	0.553	0.399	0.544
MN	-	✓	-	NT-Xent	cos	0.575	0.411	0.583

Table 1: Ablation study of design choices on the coarse-grained QBV setting. *Dual* refers to using shared or separate encoders for reference sounds and imitations; *Supervised Pre-training* indicates whether the encoders were pre-trained on the class labels in *AudioSet* and/or *VimSketch* (vocal imitations only); *Loss* indicates which loss was used for contrastive fine-tuning ('-' in this column means no training on reference–imitation pairs); *Similarity* of two embeddings was either measured with cosine similarity (*cos*) or with an *FNN*.

Model	**MRR**	**MR@1**	**MR@2**
CR-IMINET*	0.348 ± 0.03	-	-
TL-IMINET*	0.325 ± 0.03	-	-
M-VGGish	0.228 ± 0.016	0.118 ± 0.018	0.182 ± 0.018
2DFT of CQT	0.309 ± 0.021	0.169 ± 0.016	0.268 ± 0.025
MN (ours)	**0.631** ± 0.027	**0.479** ± 0.031	**0.646** ± 0.034

Table 2: Results for coarse-grained evaluation on VimSketch as described in Section 4.1; ranges give the standard deviation across the ten folds. (*Results taken from [9])

Model	**MRR**	**MR@1**	**MR@2**
TL-IMINET*	0.356	0.151	0.278
M-VGGish	0.416	0.212	0.364
2DFT of CQT	0.489	0.293	0.451
MN (ours)	0.476	0.278	0.449
MN (ours)[†]	**0.513**	**0.313**	**0.493**

Table 3: Results for the fine-grained evaluation on VocalImitationSet as described in Section 4.2. * denotes results taken from [9] and [†] indicates that supervised pre-training on vocal imitations is used.

and row 3 in Table 2). This confirms our hypothesis that MN is the stronger choice for the dual encoder setup.

Interestingly, fine-tuning the AudioSet pre-trained MN on vocal imitations (as described in Section 5.2) resulted in an MRR increase of more than 0.2 (from 0.295 to 0.508) without any contrastive training involved (compare the first rows in section 1 & 2 of Table 1).

TL- and CR-IMINET used an FNN with a single output and a Binary Cross Entropy (BCE) loss to learn the similarity between two embeddings. We tried the same with our architecture, which only resulted in a relatively small improvement (from 0.295 in row 1 to 0.354 MRR in row 2 in Table 1). Replacing the FNN with cosine similarity improved the MRR further to 0.439 (row 3 of Table 1), indicating that using the FNN head is not beneficial.

Replacing the BCE loss with the NT-Xent loss increased the performance substantially, e.g., from 0.439 to 0.635 MRR when no supervised training on vocal imitations was used (compare row 3 in section 1 and row 2 in section 3) and from 0.582 to 0.614 with supervised training on vocal imitations (compare row 2 in section 2 and row 1 in section 3). This is likely because the NT-Xent loss relies on multiple negative examples in each update.

Interestingly, supervised pre-training with vocal imitations decreased performance in combination with pre-training on AudioSet, NT-Xent loss, and cosine similarity (compare rows 1 & 2 in section 3). This indicates that AudioSet pre-training is more beneficial for coarse-grained retrieval, whereas model parameters additionally pre-trained on vocal imitations in a supervised fashion allow a more fine-grained distinction (as demonstrated by the results for fine-grained QBV in Table 3).

Using a shared embedding network for vocal imitations and reference recordings led to a performance drop (see Section 4). This is consistent with results reported by Zhang et al. [6], who also suggest that specialized encoders for the two domains are better suited for feature extraction.

Overall, the results indicate that the combination of pre-training on AudioSet with two independent embedding networks and contrastive training with NT-Xent loss enhances the retrieval accuracy of QBV systems.

6. CONCLUSION

This paper proposes a Query-by-Vocal Imitation system that improves upon previous approaches by integrating a modern, efficient CNN, pre-trained on large-scale AudioSet, in a dual encoder setup. The encoders are fine-tuned using contrastive learning with an adapted NT-Xent loss, aligning vocal imitations with their reference recordings in a shared embedding space. Our results demonstrate that the proposed system substantially enhances retrieval performance, establishing a new state of the art on both coarse- and fine-grained QBV tasks. Unlike previous deep learning-based solutions, the presented system clearly outperforms manually extracted features. We believe that our proposed system represents a step forward towards integrating QBV into sound search engines, ultimately making it easier and more intuitive to search for sounds.

7. ACKNOWLEDGMENT

The LIT AI Lab is supported by the Federal State of Upper Austria. Gerhard Widmer's work is supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme, grant agreement No 101019375 (Whither Music?).

8. REFERENCES

[1] A. S. Koepke, A. Oncescu, J. F. Henriques, Z. Akata, and S. Albanie, “Audio retrieval with natural language queries: A benchmark study,” *IEEE Transactions on Multimedia*, vol. 25, pp. 2675–2685, 2023.

[2] H. Xie, O. Räsänen, K. Drossos, and T. Virtanen, “Unsupervised audio-caption aligning learns correspondences between individual sound events and textual phrases,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2022.

[3] P. Primus and G. Widmer, “Fusing audio and metadata embeddings improves language-based audio retrieval,” in *Proceedings of the European Signal Processing Conference, (EUSIPCO), Lyon, France*, 2023.

[4] P. Primus, K. Koutini, and G. Widmer, “Advancing natural-language based audio retrieval with PaSST and large audio-caption data sets,” in *Proceedings of the Workshop on Detection and Classification of Acoustic Scenes and Events, (DCASE), Helsinki, Finland*, 2023.

[5] Y. Zhang, J. Hu, Y. Zhang, B. Pardo, and Z. Duan, “Vroom!: A search engine for sounds by vocal imitation queries,” in *Proceedings of the Conference on Human Information Interaction and Retrieval, (CHIIR), Vancouver, BC, Canada*, 2020.

[6] Y. Zhang, B. Pardo, and Z. Duan, “Siamese style convolutional neural networks for sound search by vocal imitation,” *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 27, no. 2, pp. 429–441, 2019.

[7] B. Kim and B. Pardo, “Improving content-based audio retrieval by vocal imitation feedback,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, (ICASSP), Brighton, United Kingdom*, 2019.

[8] F. Pishdadian, P. Seetharaman, B. Kim, and B. Pardo, “Classifying non-speech vocals: Deep vs signal processing representations,” in *Proceedings of the Workshop on Detection and Classification of Acoustic Scenes and Events, (DCASE), NY, USA*, 2019.

[9] B. Kim, M. Ghei, B. Pardo, and Z. Duan, “Vocal imitation set: a dataset of vocally imitated sound events using the audioset ontology,” in *Proceedings of the Workshop on Detection and Classification of Acoustic Scenes and Events, (DCASE), Surrey, UK*, 2018.

[10] M. M. Zloof, “Query-by-example: A data base language,” *IBM Syst. J.*, vol. 16, no. 4, pp. 324–343, 1977.

[11] T. Bertin-Mahieux and D. P. W. Ellis, “Large-scale cover song recognition using hashed chroma landmarks,” in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, (WASPAA), New Paltz, NY, USA*, 2011.

[12] A. Kapur, M. Benning, and G. Tzanetakis, “Query-by-beat-boxing: Music retrieval for the DJ,” in *Proceedings of the International Conference on Music Information Retrieval, (ISMIR), Barcelona, Spain*, 2004.

[13] A. Ghias, J. Logan, D. Chamberlin, and B. C. Smith, “Query by humming: Musical information retrieval in an audio database,” in *Proceedings of the Third International Conference on Multimedia, (ACM), San Francisco, CA, USA*, 1995.

[14] “Query-by-humming applied by Google LLC,” https://research.google/blog/the-machine-learning-behind-hum-to-search/, accessed: 2024-07-03.

[15] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. W. Wilson, “CNN architectures for large-scale audio classification,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, (ICASSP), New Orleans, LA, USA*, 2017.

[16] M. Cartwright and B. Pardo, “Vocalsketch: Vocally imitating audio concepts,” in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, (CHI), Seoul, Korea (South)*, 2015.

[17] A. Howard, R. Pang, H. Adam, Q. V. Le, M. Sandler, B. Chen, W. Wang, L. Chen, M. Tan, G. Chu, V. Vasudevan, and Y. Zhu, “Searching for mobilenetv3,” in *Proceedings of the International Conference on Computer Vision, (ICCV), Seoul, Korea (South)*, 2019.

[18] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, “Audio set: An ontology and human-labeled dataset for audio events,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, (ICASSP), New Orleans, LA, USA*, 2017.

[19] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” in *NIPS Deep Learning and Representation Learning Workshop, Montreal, Canada*, 2015.

[20] F. Schmid, K. Koutini, and G. Widmer, “Efficient large-scale audio tagging via transformer-to-cnn knowledge distillation,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) , Rhodes Island, Greece*, 2023.

[21] F. Schmid, K. Koutini, and G. Widmer, “Low-complexity audio embedding extractors,” in *European Signal Processing Conference, (EUSIPCO), Helsinki, Finland*, 2023.

[22] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, “A simple framework for contrastive learning of visual representations,” in *Proceedings of the 37th International Conference on Machine Learning, (ICML), Virtual Event*, 2020.

[23] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proceedings of the 3rd International Conference on Learning Representations, (ICLR), San Diego, CA, USA*, 2015.

[24] B. Kim, S. Yang, J. Kim, H. Park, J. Lee, and S. Chang, “Domain generalization with relaxed instance frequency-wise normalization for multi-device acoustic scene classification,” in *Proceedings of the 23rd Annual Conference of the International Speech Communication Association, (Interspeech), Incheon, Korea (South)*, 2022.

[25] D. R. Radev, H. Qi, H. Wu, and W. Fan, “Evaluating web-based question answering systems,” in *Proceedings of the Third International Conference on Language Resources and Evaluation, (LREC), Las Palmas, Canary Islands, Spain*, 2002.

DOES PAID CROWDSOURCING STILL PAY OFF? SIFTING THROUGH ANNOTATOR NOISE IN CROWDSOURCED AUDIO LABELS

Manu Harju, Irene Martín-Morató, Annamaria Mesaros

Signal Processing Research Centre, Tampere University, Finland
{manu.harju, irene.martinmorato, annamaria.mesaros}@tuni.fi

ABSTRACT

Paid crowdsourcing has emerged as a popular method for annotating diverse data types such as images, text, and audio. However, the amount of carelessly working annotators has increased as platforms have become more popular, leading to an influx of spam workers that answer at random, which renders the platforms unusable. This paper documents our attempt to annotate the DESED dataset using Amazon's Mechanical Turk, and failing to obtain any useful data after two attempts. Our observations reveal that while the number of workers performing the tasks has increased since 2021, the quality of obtained labels has declined considerably. After successful trials for annotating audio data in 2021 and 2022, in 2024 the same user interface annotation setup predominantly attracted spammers. Given the consistent task setup and similarity to previous attempts, it remains unclear whether the workers are inherently subpar or if they are intentionally exploiting the platform. The bottom line is that despite spending a considerable amount of money on it, we obtained no usable data.

Index Terms— Data annotation, crowdsourcing

1. INTRODUCTION

Crowdsourcing is a collaborative online process where a group of individuals with different skills, knowledge, and backgrounds is participating to work on some *task*. Tasks are usually surveys, data annotations, description collections, or other such assignments which are difficult for computers but easy for humans [1]. Amazon Mechanical Turk (AMT) uses the term Human Intelligence Task (HIT) for a single annotation/answer. Crowdsourcing involves two key roles: *requesters*, who create data-collection tasks, and *workers*, who complete those tasks. In paid crowdsourcing, requesters compensate workers for their completed assignments. One benefit of using paid crowdsourcing platforms is their vast pool of workers. However, since the work is done by humans with varying abilities and backgrounds, the crowdsourced results are likely to contain some amount of errors. Some errors are simply mistakes, but there are also workers aiming to collect the task rewards without caring much about their work.

The quality of the crowdsourcing results can be improved by (1) taking more control of the data collection process itself, and (2) using different postprocessing and aggregation methods. The former means checking the correctness of some part of the annotations, and rejecting incorrect ones and possibly banning the workers from taking more tasks. In case of a label assigning task, the latter can be done e.g. by directly optimizing the labels or through estimating the reliabilities of the individual annotators. The study in [1] presents a good overview of different aggregation methods. In practice, the two approaches should be used together, but often the purpose of using crowdsourcing is lost if keeping the annotation process clean requires too much effort.

The general setting in the crowdsourcing platforms makes the workers to do *invisible labour*, meaning that part of the time spent on the platform does not generate any income. Invisible labour includes e.g. rejected work, finding new tasks, interacting with requesters, and managing payments [2]. A study from 2018 reports that the average requester on Amazon Mechanical Turk paid $11/hour. However, lower-paying requesters were publishing more work, and as an effect the median wage for workers was approximately $2/hour [3]. Due to the factors explained above, working on microtasks can be difficult to make profitable.

There has been some development of guidelines for requesters on how to make their tasks ethical, e.g by having clear instructions and examples of good answers for the task, and reasonable reward for the tasks. Furthermore, Hiippala *et.al.* argue that human errors should not be a reason for rejection [2]. This creates a problem for the requester: how to recognize when something is a human error and not a bad-faith answer? To be sure to stay on the fair side, the requester should only reject the most obvious cases, e.g. tasks done in too short time. This, in turn, opens up the opportunity for the workers to exploit the requesters by doing the task carelessly or simply bypassing the task and instead providing a response that *seems* correct. The study in [4] shows that the amount of bad survey data has risen from 2% in 2013 and 5% in 2018 to almost 89% in 2022; the authors bring up the same question of how to distinguish a bad-faith answer. They also note that the workers were likely to either co-operate closely with each other or use multiple accounts, as some of the answers were too similar.

In audio, paid crowdsourcing has been used for creating datasets of speech transcriptions [5], audio captions [6], positive and negative audio-caption pairings [7]. However, hearing and classification of sounds are subjective, and e.g. the annotation context and the worker's personal background affects the recognized sounds [8]. Furthermore, requesters can only recommend but cannot control the environment and equipment the workers are using for the tasks, making the distinction between a human error and a bad-faith answer even more trickier.

This paper documents the efforts we made in 2024 to annotate part of the DESED [9] data for the DCASE 2024 Sound Event Detection Task. Our previous work has repeatedly shown that it is possible to obtain reliable annotations for sound events. We started with a study using synthetic data [10]; as the process was shown to work, we moved on to annotate real data [11]. Unfortunately, it seems that the process is no longer working as expected. The contributions of

This work was supported by Academy of Finland grant 332063 "Teaching machines to listen".

the paper are: (1) we analyze the quality of annotations obtained through paid crowdsourcing, observing that it has decreased considerably in a few years, and (2) we show that multi-annotator competence estimation (MACE) [12] is robust against bad-faith annotators, even in large quantities.

2. COLLECTING THE DATA

2.1. Annotation setup

In all the annotation experiments in this paper we followed the procedure presented in [10]. The main idea was to break down the complicated task of annotating onset and offset times beside the class labels of sound events into a simple tagging task of highly overlapping sound clips. Afterwards, the temporally weak annotations could be aggregated with the temporal information. The sound clips used in the experiments were 10-second clips cut out from longer pieces of audio. The start times for the clips were increased one second at a time, such that two consecutive clips have nine seconds of overlapping audio. Each clip was annotated by multiple workers, 5 in the previous work, and 3 in the current experiments. We opted for the lower number now due to the high number of clips to annotate and therefore high cost. As a consequence of the overlap, each one-second segment of audio was included in a total of 50 annotation tasks (30 in 2024). Each annotator's competence value was estimated using MACE [12], and the labels were reconstructed by taking weighted averages over all the opinions that included each one-second segment using the competences as weights [11].

For the first experiment in 2021, the audio was generated by using the isolated sound events from UrbanSound8k [13]. The events were sampled from six classes, and the synthesized dataset consisted of 20 3-minute long files [11]. For the following experiments for MAESTRO Real [14] in 2021 and 2022 we used data recorded from five different scenes of the TUT Acoustic Scenes 2016 dataset [15]; for each scene we used six event classes. Due to some overlap in the classes, the total number of classes of the resulting dataset is 17, but in the HITs the tasks were presented per acoustic scene, i.e. with only six classes to tag. Finally in January 2024, we aimed to annotate 556 files for the evaluation set of the Sound Event Detection task in DCASE 2024 Challenge. For this last annotation task, the target annotation length was 10 seconds; in order to cover this length, due to the annotation method explained above, the source files were 28 seconds long, including 9 extra seconds on each side of the target segment. Furthermore, the number of event classes was ten instead of the six used in previous experiments.

We verified that using three annotators per file instead of five is sufficient by sampling annotations using the data from MAESTRO Real experiments. Using only three randomly selected annotators per file gave similar results as the reconstruction based on five annotators.

2.2. Task description

The task layout used to collect the annotations contained an audio player, a short list of instructions, and a selector for the event classes. The instructions advised doing the experiment in a quiet environment and with good quality headphones. It was mentioned that the annotators could playback the audio as many times they wanted. The annotators were asked to select all the sound event classes they can recognize in the clip from the given list.

In all experiments the files were divided into 15 different batches based on their start time. The first batch contained all the clips with start times 0, 15, 30, . . . , the second batch with start times 1, 16, 31, . . . , and so on. By this construction, the gap between two clips in a batch is always at least 15 seconds.

We required workers to have at least 1000 completed HITs and at least 90% approval rate. In practice, we accepted almost all annotations. The annotations completed in shorter time than the sound clip were taken into closer inspection, and the ones tagging clearly incorrect labels were rejected. However, the rejected tasks annotators were not banned from taking more tasks. One thing we noticed in the last experiment was that the workers deduced this and simply spent more time on the task such that these "too fast" annotations were not anymore present in the later batches.

2.3. Two attempts

In the first DESED annotation (DESED/A1), we introduced fields for the annotator confidence: for any positive label assigned, the annotator had to specify how confident they were about the label. The confidence was given on a six-step scale from 50% to 100% with 10% increments. The scope was to study the relationship between estimated competence and self-evaluated confidence of the annotators.

After the data collection we noticed that the competence estimation resulted in a very skewed distribution, where most of the competence values were centered close to zero. Furthermore, the aggregated labels for most of the classes did not agree very well with the reference annotation[1], and aggregating the annotations using the previously used method resulted in useless data. There was a large number of annotators doing only a few tasks, hinting that the task setup was too complicated and driving the workers away. The number of available HITs was approximately three-fold compared to the earlier experiments, but the highest number of files annotated by a single worker in DESED/A1 was 112.

We do not know what caused the high number of bad annotations. Based on the task setup, there are two possible factors. First, the number of classes was increased from six to ten, making an individual task more complex. Second, the annotators had to answer the question about confidence for each positive label, which adds to the annotators' work load. We also hypothesized that the reason for such a unusual competence distribution was that the data was too sparse for MACE to handle, due to the high number of annotators doing few HITs. We decided to repeat the process without the confidence question. For the second DESED annotation (DESED/A2) we reverted to the basic task layout to see if there was any difference without the question about confidence. Unfortunately, in terms of label quality, we ended up with similar results as in DESED/A1.

3. SIFTING THROUGH THE DATA

3.1. Analysis of the outcome

We started the analysis of DESED/A1 with the standard approach by estimating the annotators' competence values using MACE. The competences can vary from 0 to 1, and according to MACE the vast majority of the annotators had extremely low competence values: the median competence was 0.09 and the fraction of annotators with a competence value smaller than 0.01 was over 19%. Figure 1 shows the histograms of the competence values in both experiments. In DESED/A1 the amount of workers annotating at most 5 files was 51%; this number decreased to 36% in DESED/A2. However, the

[1] Reference annotation available as manually annotated strong labels [9]

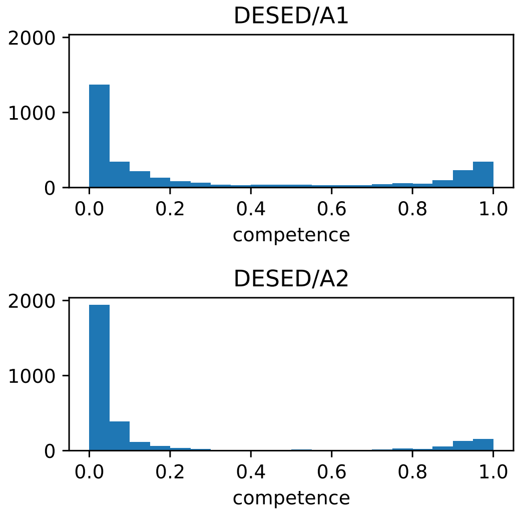


Figure 1: Histograms of the estimated competences.

Scene	Date	#Clips	#Workers	Acc. workers
Synthetic	3/2021	3420	680	680
City center	6/2021	3544	717	1154
Residential area	6/2021	3429	861	1517
Cafe/restaurant	9/2022	3273	1554	2870
Grocery store	9/2022	2840	1509	3450
Metro station	9/2022	3418	1641	3832
DESED/A1	1/2024	10545	3295	6711
DESED/A2	6/2024	10545	3059	8125

Table 1: Annotation dates and numbers of individual sound clips and annotators. The last column shows the cumulative number of workers that participated in our data collections.

Scene	F-score	Precision	Recall
Synthetic	72.3	89.3	62.8
City center	50.4	60.9	45.6
Residential area	51.0	57.2	50.0
DESED/A1	37.0	50.9	31.4
DESED/A2	37.8	51.4	31.7

Table 2: Average tagging scores in different experiments.

,

competence distribution in DESED/A2 was even more skewed than in DESED/A1. The numbers suggest that removing the question about confidence made the task more attractive for workers, but maybe also less engaging.

The annotators often disagreed with each other. In terms of aggregation, having only three opinions per file instead of five accentuated the problem caused by disagreements. 62% of the clips in DESED/A1 had completely disjoint class labels from the three annotators, and this number increased to 87% in DESED/A2. We observed that annotators also disagreed with themselves: there were 20 annotators who annotated the same file in both DESED/A1 and DESED/A2, and in 15 of the cases they assigned completely different sets of labels. The inconsistencies can be due to changes in the circumstances, but either the sounds are very hard to recognize, or the workers did not perform the task genuinely. Nevertheless, these findings illustrate the randomness of the annotator behavior overall.

Table 1 shows the dataset sizes and numbers of workers, as well as the time of the data collection. Due to the long gap between the last MAESTRO Real annotation and DESED/A1, it is understandable that 87% of the worker accounts were new in our experiments. However, between the two DESED experiments there was less than a five months gap, and still almost half of the annotator accounts in DESED/A2 were completely new to our tasks.

3.2. Tagging precision and MACE

For some of the scenes there exist temporally strong labels. We converted the available labels into tags of the annotated clips to measure each annotator's tagging performance. The tagging performances over the workers in different scenes are shown in Table 2. To check the overall quality of the answers, we also calculate the average precision over HITs. With this, precision in DESED/A1 and A2 drops to 43.2 and 43.9, respectively. This indicates that the workers completing more tasks are not producing the better labels.

The worker competence is computed based on the tagging task, and we expect a connection between the competence and tagging performance. In Fig. 2 we show the precision on the individual and combined experiments. The annotators are divided into equally-sized groups based on their competence values, with the bin borders marked on the x-axis. The competence quantiles are very skewed: in DESED/A1 3/5 of the workers have a competence less than 0.17, and in DESED/A2 4/5 of the workers have competence less than 0.15. Combining the data from the two flattens the competence distribution, but adds a few outliers in the plot. These results indicate that MACE is still able to identify the better performing workers despite the vast amount of noise in the annotation. Furthermore, combining the data seems to improve the MACE output. Unfortunately we do not have enough reliable annotations even when the experiments are combined.

3.3. Comparing aggregated labels against reference data

We compare the reconstructed soft labels to the reference data using macro soft F-score [16] to avoid the problem of choosing the threshold value for binarizing the data. Table 3 shows the F-scores for the scenes we have a reference annotation available. The scores for both DESED experiments are similar to each other, but also extremely low. Furthermore, when the annotation data is combined from the two experiments, the standard method results in worse labels than either of the experiments alone. Table 3 also includes the average competence C_{avg} evaluated using MACE for each annotation set. The average competence is not telling the whole truth, as the weighting is in practice determined by the differences between the competences related to a single segment. This can also be seen in the combined case, where the average competence is as high as 0.58.

For further analysis, we can inject the reference annotation into the competence estimation along with the collected labels to obtain a competence value C_{ref} for the reference labels. If the reference labels mostly agree with the annotations, the annotators and the reference should have a high competence. Similarly, if the reference labels are mostly different from the annotated labels, MACE interprets the reference as an annotator submitting random labels, resulting in a low competence value. Combining the data from the two experiments improves the MACE results in terms of higher C_{avg} and C_{ref}, but does not change much the competence distribution.

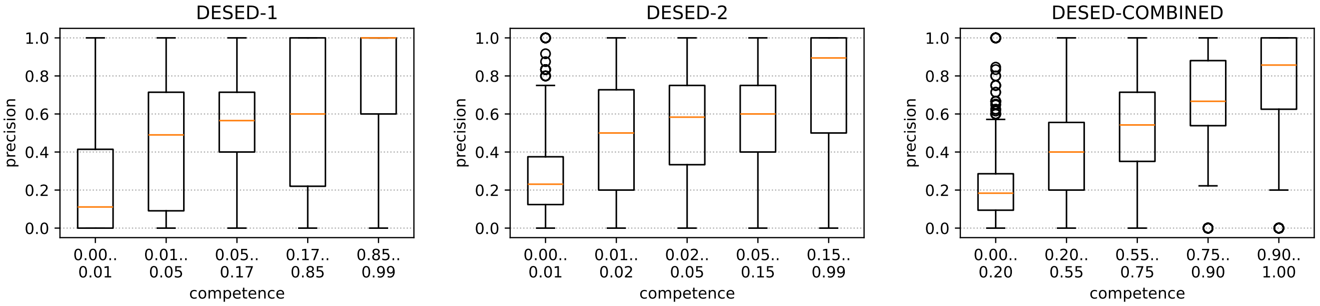


Figure 2: Tagging precision for DESED/A1, DESED/A2, and the combined data. Workers are grouped by their competence values into equally-sized groups. The skewness of the competence value distributions in the two DESED experiments can be seen in the bin borders.

Scene	F_M	C_{avg}	C_{ref}
Synthetic	63.8	0.73	0.89
City center	45.4	0.43	0.68
Residential area	39.3	0.53	0.57
Cafe/restaurant	-	0.43	-
Grocery store	-	0.42	-
Metro station	-	0.34	-
DESED/A1	31.2	0.31	0.35
DESED/A2	31.0	0.17	0.36
DESED/A1 + A2	29.1	0.58	0.61

Table 3: Soft macro F-scores F_M of the reconstructed sound event labels, average competences of the annotators C_{avg}, and the reference annotation competences C_{ref} when injected into the datasets. For the three scenes without a reference annotation available, only the average competence is shown.

3.4. Competence value clamping

The reconstructed soft value for a segment is a weighted average of the annotators labels, using the competence values as weights. For DESED, MACE estimated a majority of the annotators to have a competence value close to zero; this might cause some instabilities in the label reconstruction, if all the annotators for a given segment have very low competences. Furthermore, MACE uses a stochastic method, resulting in fluctuation in the output values. However, the small differences in the competence values can result in unexpectedly large differences when weighting the labels, while, intuitively, if the annotators are equally bad, they should have equal weights.

As an additional experiment, we assume that all low-competent annotators are equally bad, and clamp the competence values of the lowest ranking annotators to a small fixed value. We use 10^{-4} as the competence value, and set it as the competence of the worst 50% and 75% of the annotators. Table 4 shows the comparison between the labels generated from the original competence values and labels generated from the partially clamped competences, as well as using equal weights for all annotators. The standard method is better than not using any weighting, but using the MACE-estimated competences results in a lower F-score than resetting the competences of the lowest ranking annotators, to different degree for DESED/A1 and DESED/A2; furthermore, while combining the DESED/A1 and DESED/A2 annotations shows no benefit with the standard procedure, resetting the lowest competences to the same small value produces the best scoring soft labels. While having more data in DESED/A1+A2 results in a wider distribution of competences and better correspondence with precision, according to Fig. 2, the underlying problem of bad quality labels remains unchanged.

Scene	Original	R50	R75	EQ
DESED/A1	31.2	32.4	34.3	21.8
DESED/A2	31.0	31.9	32.8	22.0
DESED/A1 + A2	29.1	33.8	34.5	21.8

Table 4: Soft macro F-scores for the reconstructed labels and the effect of competence value resetting. In R50 and R75, the lowest-competent 50% and 75% of the annotators, respectively, have competence value reset to 10^{-4}. EQ denotes equal competences.

3.5. Discussion

It is difficult to draw the border between a bad-faith answer and a simple mistake, especially when the task involves human hearing. The problem of bad-quality answers is not platform specific [17], and hence not limited to our experience in using AMT. At the time of our first annotation experiments, there were already discussions about the data quality in paid crowdsourcing [18, 19, 20]. However, in our previous annotation experiments, the amount of low quality work did not hamper significantly the end result quality, unlike now. Based on this work, it seems that MACE is able to identify the annotators producing good quality labels. The problem arises, though, when there are no reliable annotations for a segment, in which case the output annotation ends up having noisy labels.

We speculated that asking annotators' confidence made the annotation somehow annoying or more difficult, causing workers to abandon it after a few HITs. Removing the confidence question indeed decreased the number of annotators who only completed a few HITs and increased the average task count of the workers, but it did not improve the label quality.

4. CONCLUSIONS

This paper presented a detailed analysis of the labels produced by a crowdsourcing process. The approach was to collect temporally strong labels by dividing the work into simpler subtasks of weak labeling, a method previously proven to work. Our conclusion is that the quality of crowdsourced work has worsened considerably, rendering the process unusable. It is hard to pinpoint the reason for this decrease in quality, with potential causes being the influx of workers gaming and exploiting the process, the perceived unfair difficulty/payment ratio of the task, etc. It may be possible to collect sufficiently good labels by simply using more workers, but the process gets prohibitively expensive, driving researchers to return to doing manual annotation themselves.

5. REFERENCES

[1] Y. Zheng, G. Li, Y. Li, C. Shan, and R. Cheng, "Truth inference in crowdsourcing: is the problem solved?" *Proc. VLDB Endow.*, vol. 10, no. 5, pp. 541—552, 2017.

[2] T. Hiippala, H. Hotti, and R. Suviranta, "Developing a tool for fair and reproducible use of paid crowdsourcing in the digital humanities," in *Proceedings of the 6th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*. Gyeongju, Republic of Korea: International Conference on Computational Linguistics, 2022, pp. 7–12.

[3] K. Hara, A. Adams, K. Milland, S. Savage, C. Callison-Burch, and J. P. Bigham, "A data-driven analysis of workers' earnings on amazon mechanical turk," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, ser. CHI '18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 1—14.

[4] C. C. Marshall, P. S. Goguladinne, M. Maheshwari, A. Sathe, and F. M. Shipman, "Who broke amazon mechanical turk? an analysis of crowdsourcing data quality over time," in *Proceedings of the 15th ACM Web Science Conference 2023*, New York, NY, USA, 2023, pp. 335–345.

[5] N. Pavlichenko, I. Stelmakh, and D. Ustalov, "Crowdspeech and vox DIY: Benchmark dataset for crowdsourced audio transcription," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.

[6] K. Drossos, S. Lipping, and T. Virtanen, "Clotho: an audio captioning dataset," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 736–740.

[7] H. Xie, K. Khorrami, O. Räsänen, and T. Virtanen, "Crowdsourcing and evaluating text-based audio retrieval relevances," in *Proceedings of the 8th Detection and Classification of Acoustic Scenes and Events 2023 Workshop (DCASE2023)*, M. Fuentes, T. Heittola, K. Imoto, A. Mesaros, A. Politis, R. Serizel, and T. Virtanen, Eds., 2023, pp. 226–230.

[8] C. Guastavino, *Everyday Sound Categorization*. Springer International Publishing, 2018, pp. 183–213.

[9] R. Serizel, N. Turpault, A. Shah, and J. Salamon, "Sound event detection in synthetic domestic environments," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, 2020.

[10] I. Martín-Morató and A. Mesaros, "Strong labeling of sound events using crowdsourced weak labels and annotator competence estimation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 902–914, 2023.

[11] I. Martín-Morató, M. Harju, and A. Mesaros, "Crowdsourcing strong labels for sound event detection," in *2021 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2021, pp. 246–250.

[12] D. Hovy, T. Berg-Kirkpatrick, A. Vaswani, and E. Hovy, "Learning whom to trust with MACE," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, L. Vanderwende, H. Daumé III, and K. Kirchhoff, Eds., Atlanta, Georgia, 2013, pp. 1120–1130.

[13] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *Proceedings of the 22nd ACM International Conference on Multimedia*, ser. MM '14. New York, NY, USA: Association for Computing Machinery, 2014, pp. 1041—1044.

[14] I. Martín-Morató, M. H. Harju, and A. Mesaros, "MAESTRO Real - Multi-Annotator Estimated Strong Labels," Feb. 2023. [Online]. Available: https://doi.org/10.5281/zenodo.7244360

[15] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *2016 24th European Signal Processing Conference (EUSIPCO)*, 2016, pp. 1128–1132.

[16] M. Harju and A. Mesaros, "Evaluating classification systems against soft labels with fuzzy precision and recall," in *Proceedings of the 8th Detection and Classification of Acoustic Scenes and Events 2023 Workshop (DCASE2023)*, 2023, pp. 46–50.

[17] C. Schild, L. Lilleholt, and I. Zettler, "Behavior in cheating paradigms is linked to overall approval rates of crowdworkers," *Journal of Behavioral Decision Making*, vol. 34, no. 2, pp. 157–166, 2021.

[18] R. Kennedy, S. Clifford, T. Burleigh, P. D. Waggoner, R. Jewell, and N. J. G. Winter, "The shape of and solutions to the MTurk quality crisis," *Political Science Research and Methods*, vol. 8, no. 4, pp. 614—629, 2020.

[19] M. Chmielewski and S. C. Kucker, "An MTurk crisis? Shifts in data quality and the impact on study results," *Social Psychological and Personality Science*, vol. 11, no. 4, pp. 464–473, 2020.

[20] M. Dupuis, K. Renaud, and R. Searle, "Crowdsourcing quality concerns: An examination of amazon's mechanical turk," in *Proceedings of the 23rd Annual Conference on Information Technology Education*, ser. SIGITE '22, New York, NY, USA, 2022, pp. 127—129.

ENCLAP++: ANALYZING THE ENCLAP FRAMEWORK FOR OPTIMIZING AUTOMATED AUDIO CAPTIONING PERFORMANCE

Jaeyeon Kim[1,2], *Minjeong Jeon*[2], *Jaeyoon Jung*[2,3], *Sang Hoon Woo*[4], *Jinjoo Lee*[2],

[1] Seoul National University, Seoul, Republic of Korea, jaeyeonkim99@snu.ac.kr
[2] MAUM AI Inc., Seongnam, Republic of Korea, {mjjeon, jyjung, jjl}@maum.ai
[3] Soongsil University, Seoul, Republic of Korea
[4] Independent Researcher, tonyswoo@gmail.com

ABSTRACT

In this work, we aim to analyze and optimize the EnCLAP framework, a state-of-the-art model in automated audio captioning. We investigate the impact of modifying the acoustic encoder components, explore pretraining with different dataset scales, and study the effectiveness of a reranking scheme. Through extensive experimentation and quantitative analysis of generated captions, we develop EnCLAP++, an enhanced version that significantly surpasses the original.

Index Terms— Automated audio captioning, language-based audio retrieval, neural audio codec, audio-text joint embedding

1. INTRODUCTION

Automated audio captioning (AAC), a cross-modal translation involving transcribing audio signals into concise and meaningful natural language descriptions [1], remains a particularly challenging task with a substantial performance gap between human and machine. One significant contributor to the performance gap can be attributed to the intrinsic complexity of the task, as distinguishing between various sound events, especially between similar and ambiguous ones, requires extensive real-world knowledge. Furthermore, the scarcity of high-quality data, with the most widely used datasets, AudioCaps [2] and Clotho [3] containing only 50K and 20K captions, respectively, poses an additional challenge. To address these challenges, prior studies have employed pretrained audio encoders trained on audio classification tasks [4, 5, 6], leveraged the text generation capabilities of pretrained language models like GPT-2 [7, 8, 9] and BART [10, 11], and incorporated auxiliary loss terms, including keyword prediction loss [12] or sentence embedding loss [13], to improve the semantic quality of captions and provide additional training signal.

Building on the previous line of research, Kim *et al.* [14] proposed the EnCLAP framework which integrates a set of pretrained models with an auxiliary training task. Specifically, EnCLAP utilizes two acoustic feature encoders, EnCodec [15] and CLAP [16], to generate timestep-level and sequence-level representation of the input audio sequence, respectively. EnCLAP utilizes pretrained BART as the caption decoder to leverage these features and generate captions. Furthermore, Kim *et al.* also introduced masked codec modeling (MCM), an auxiliary task which involves masking a part of the input codec sequence and predicting it, to enhance the acoustic awareness of the caption decoder. The caption decoder was trained jointly using cross-entropy loss for caption generation and MCM loss. The combination of these approaches allowed EnCLAP to achieve state-of-the-art performance on the AudioCaps dataset.

Although EnCLAP exhibits impressive performance, the study by Kim *et al.* lacks sufficient experimental evaluation for determining the optimal models for the model components. Notably, the authors do not investigate alternative sequence-level acoustic features beyond CLAP. Furthermore, for timestep-level acoustic features, while they demonstrate that discrete codec input outperforms continuous input, their analysis is restricted to a single setup using EnCodec, without exploring other options or configurations. Additionally, Kim *et al.* acknowledge the issue of overfitting in larger model variants but do not investigate the use of large-scale weakly-labeled datasets [17, 6], which contain noisy and model-generated captions. Therefore, the EnCLAP framework has potential for further optimization.

In this work, we extend and optimize the EnCLAP framework through a comprehensive examination of its components. We explore alternative acoustic feature encoder components and assess their efficacy. We also investigate the impact of large-scale training incorporating weakly-labeled datasets on the framework's performance. Furthermore, we adopt a sampling-and-reranking approach [6] as an alternative to beam search decoding and evaluate its effectiveness. Finally, we conduct a qualitative analysis of the generated captions to examine the effects of each component on the outputs. Based on our findings, we present EnCLAP++, an improved version of the EnCLAP model that achieved second place in the DCASE2024 Challenge Task6. Figure 1 provides an overview of EnCLAP++.

2. EXPERIMENTAL DESIGN

2.1. Timestep-level Acoustic Embedding

Neural audio codecs are autoencoder models designed to encode waveforms into sequences of discrete codes. Recent advancements [18, 15, 19] typically employ residual vector quantization (RVQ) for compression, utilizing multiple codebooks to quantize the residuals of preceding codebooks. Ultimately, the input waveforms are transformed into a set of parallel discrete code sequences, each of which is associated with a unique codebook. Neural audio codecs have demonstrated success as the acoustic representation format in generative audio models [20, 21, 22].

Kim *et al.* [14] demonstrate that language models achieve superior performance when used with discrete input sequences compared to continuous input sequences. However, their study does not explore the impact of different configurations within the discrete input sequence setup. To address this limitation, we conduct experiments to examine the effects of different codec settings on the model performance. Specifically, we investigate the effect of codebook size,

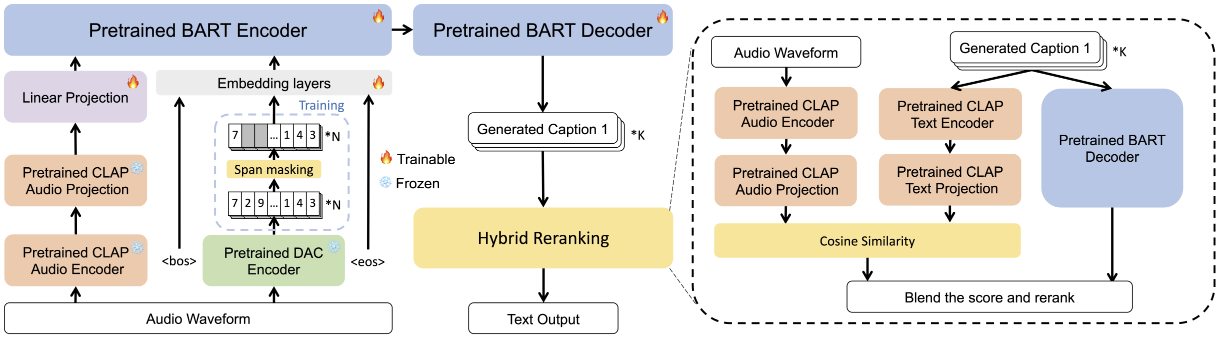


Figure 1: Overall architecture of EnCLAP++

sample rate, and codec type on the final outcome.

The original EnCLAP employed a version of EnCodec [15] that compresses a 24kHz audio signal into 16 discrete code sequences at a rate of 75Hz, with a codebook size of 1024. We experiment with two additional variants of EnCodec, which yield 8 and 32 code sequences, respectively, as well as a variant that processes 48kHz audio signal input. As for the alternative codec, we use a variant of Descript Audio Codec (DAC) [19] that closely resembles the original Encodec setup, which transforms 24kHz audio signal into 32 code sequences at a rate of 75Hz. We opted for DAC as the alternative codec due to its superior performance in audio compression, as well as downstream tasks [19, 23].

2.2. Sequence-level Acoustic Embedding

While EnCLAP employs CLAP [16] as its sequence-level acoustic feature encoder, preceding studies in audio captioning have predominantly utilized models pretrained on the AudioSet [24] dataset for audio classification task [4, 5, 6]. In this work, we investigate alternative candidates for the sequence-level acoustic encoder component. Specifically, we examine the sequence-level representation capabilities of a model pretrained on AudioSet with audio tagging task and its variants, which have gone through additional audio-text retrieval training. We compare the audio captioning performance of these models with the original CLAP setup and assess the impact of additional retrieval training on downstream performance.

For the baseline sequence-level encoder, we use ConvNext-Tiny [25] pretrained on AudioSet classification, referred to as CNext, and three of its variants that have undergone additional training on datasets of varying scales. Specifically, the three dataset configurations are: (1) Clotho [3], (2) AudioCaps [2] and Clotho, and (3) WavCaps [17], AudioCaps, and Clotho. We use m-LTM framework [26] and bge text encoder [27] for retrieval training. We assess the performance of these models against the original CLAP version.

2.3. Large-scale Pretraining

The original EnCLAP described two versions of the model, denoted as "base" and "large", based on the size of the underlying BART [10] model used. The study highlights the issue of overfitting, especially in the large variant with smaller training datasets. To mitigate this issue, we draw on the recent trend in audio captioning, which involves leveraging weakly-labeled datasets for pretraining [17, 6]. In particular, we evaluate a large-scale pretraining setup, where the model is pretrained on the WavCaps, and finetuned on Clotho, against the original EnCLAP dataset setup, where the model is pretrained on AudioCaps and finetuned on Clotho. From WavCaps, we filter out audio clips that fall outside the 1-30 second duration range, as well as overlapping clips from AudioCaps and Clotho. We evaluate both setups using both the base and large variants of our model.

2.4. Generation and Reranking

Previous works, including EnCLAP, have utilized beam search decoding for caption generation. However, Wu *et al.* [6] demonstrates that the sampling-then-reranking approach yields more diverse and informative captions. Wu *et al.* proposes two scores for candidate reranking: the encoder reranking score and the decoder reranking score. The encoder reranking score is the cosine similarity score between the input audio representation and the generated caption representation computed using a retriever model. The decoder reranking score is the log-likelihood of the generated caption given the input audio. In this study, we explore the benefits of incorporating the reranking scheme into the EnCLAP framework. Specifically, we compare the original beam search scheme against three reranking setups: encoder reranking, decoder reranking, and hybrid reranking. We use CLAP as the retriever model for computing the encoder reranking score. We perform a fluency error-based filtering before the reranking procedure, following Wu *et al.*.

For sampling, we use nucleus sampling with a probability threshold of 0.95 and a temperature of 0.5 to generate 30 candidates. For hybrid reranking, we rank the candidates by the weighted sum of the encoder reranking score and the decoder reranking score using weights of 0.6 and 0.4, respectively.

2.5. Quantitative Evaluation Metric

We adopt both widely used AAC metrics, METEOR, CIDEr, SPICE, and SPIDEr, and more recently proposed AAC metrics, SPIDEr-FL, FENSE [28], and Vocab to evaluate various aspects of the generated captions. All metrics are calculated using the aac-metrics library. METEOR is a machine translation evaluation metric, based on unigram precision and recall. CIDEr and SPICE assess the syntactic and semantic quality of the generated captions, respectively, while SPIDEr is a linear combination of them. SPIDEr-FL is SPIDEr score penalized by the fluency error. FENSE is the combination of the SentenceBERT similarity score and the fluency error penalty. Vocab shows the diversity of the vocabularies in the generated captions.

2.6. Qualitative Analysis

Although quantitative metrics provide valuable insights into relative improvements in model performance, they are inherently limited, particularly in tasks such as audio captioning, where no single objective

Table 1: Evaluation Results on Clotho. Ret refers to retrieval finetuning on the datasets listed in parentheses. CL, AC, and WC represent the Clotho, AudioCaps, and WavCaps datasets, respectively. Base and Large indicate the size of the pretrained BART model.

Model	METEOR	CIDEr	SPICE	SPIDEr	SPIDEr-FL	Vocabulary	FENSE
Timestep-level Representations							
EnCLAP-base	0.180	0.461	0.128	0.294	0.291	535	0.497
w/ EnCodec, 8 codebooks	0.178	0.444	0.127	0.286	0.283	626	0.497
w/ EnCodec, 32 codebooks	0.180	0.446	0.128	0.287	0.285	**658**	0.503
w/ EnCodec, 48khz	0.179	0.441	0.125	0.283	0.281	610	**0.505**
w/ DAC	**0.183**	**0.463**	**0.131**	**0.297**	**0.294**	589	0.504
Sequence-level Representations							
CLAP + DAC	**0.183**	**0.463**	**0.131**	**0.297**	**0.294**	589	0.504
CNext + DAC	0.175	0.426	0.120	0.273	0.269	584	0.488
CNext + Ret(CL) + DAC	0.179	0.431	0.127	0.279	0.274	**677**	0.497
CNext + Ret(CL+AC) + DAC	0.181	0.454	0.130	0.292	0.287	596	0.500
CNext + Ret(CL+AC+WC) + DAC	0.179	0.452	0.127	0.290	0.286	676	**0.508**
Large-Scale Pretraining							
Base	0.183	0.463	0.131	0.297	0.294	589	0.504
Large	0.184	0.393	0.132	0.262	0.260	571	0.480
Base + WC Pretraining	0.185	**0.470**	**0.134**	**0.302**	**0.299**	**620**	**0.505**
Large + WC Pretraining	**0.187**	0.464	0.130	0.297	0.293	576	0.500
Generation and Reranking							
Beam Search	0.185	0.470	0.134	0.302	0.299	620	0.505
Beam Search without Fluency error	0.185	0.470	0.135	0.302	0.302	619	0.511
Encoder Reranking	0.176	0.396	0.126	0.261	0.261	**915**	0.520
Decoder Reranking	0.187	0.460	0.139	0.299	0.299	608	0.506
Hybrid Reranking	**0.190**	**0.479**	**0.142**	**0.310**	**0.310**	699	**0.526**

truth exists. Thus, in addition to reporting quantitative metrics, we perform a qualitative analysis of the generated captions. Specifically, we identify the examples with the largest improvement in the evaluation metric between the baseline and the best-performing variant and manually examine the enhancement in the caption quality.

3. RESULTS AND ANALYSIS

3.1. Timestep-level Acoustic Embedding

Table 1 shows that substituting the EnCodec encoder with an alternative variant does not enhance the model's performance and, in fact, leads to incremental degradation. This indicates that changing the timestep-level feature encoder across different EnCodec models has a negligible effect on the performance in the audio captioning task. Contrastively, replacing the EnCodec encoder with the DAC encoder leads to a modest improvement in the model performance. We believe that the DAC's superior ability to preserve the information in the original audio signal contributes to the enhancement. Therefore, we adopt DAC as the timestep-level acoustic feature encoder in subsequent experiments.

3.2. Sequence-level Acoustic Embedding

As illustrated in Table 1, the model using CNext as the sequence-level acoustic encoder falls behind the CLAP variant. However, the results indicate that additional retrieval training boosts the audio captioning performance and further, increasing the dataset size narrows the performance gap relative to the CLAP variant. Nevertheless, none of the CNext variants fully surpass the CLAP variant in terms of performance. We attribute the performance gap to the fact that CLAP was trained on a much larger scale than CNext, even with additional training, which is consistent with our findings within the CNext variants. Consequently, we will proceed with the original CLAP variant in subsequent experiments.

3.3. Large-scale Pretraining

The third section of Table 1 demonstrates the effect of augmenting the pretraining dataset with a large-scale weakly-labeled dataset. Notably, our results for the original dataset setup replicate the phenomenon observed in the original EnCLAP work, where the large variant performs worse than the base variant. While variants with large-scale pretraining also exhibit this issue, the performance degradation is significantly less pronounced. Given that large-scale pretraining substantially improves the base variant, we infer that even the base variant can benefit from larger datasets. Our hypothesis is that larger datasets are necessary to fully utilize the capabilities of the large variant models.

3.4. Generation and Reranking

We investigated sampling and reranking techniques using the base variant pretrained on WavCaps from Sec 3.3. The results are presented in the last section of Table 1. Our findings indicate that encoder reranking enhances both the diversity of words and the semantic content of the generated captions. However, this improvement in semantic quality comes at the expense of syntactic quality. In contrast, decoder reranking alone yields results comparable to beam search, while when encoder and decoder reranking are combined, there is a significant improvement in semantic quality without any degradation in syntactic quality.

3.5. Qualitative Analysis

Timestep-level Acoustic Embedding. The variant without DAC tends to focus on the most prominent event in a clip, but frequently overlooks background and supplementary acoustic events. This shortcoming can be attributed to the inherent constraint of relying on a single vector to represent the entire clip, which can lead to a loss

Table 2: Example of the generated captions.

	Timestep-level Representations	
w/o DAC	w/ DAC	Ground Truth
A person walks on a hard surface at a constant pace	A person is walking on a hard surface while birds are chirping	A person walking down a beach boardwalk with seagulls squawking overhead and people chatting in the background near the end
A woman is speaking over an intercom to a crowd of people	A man is speaking on a radio with people talking in the background	A man is talking on a radio with singing in the background
A door creaks as it is opened and closed several times	A person is walking on a wooden floor while birds chirp in the background	Someone walking slowly as birds chirp in the background
	Sequence-level Representations	
w/o CLAP	w/ CLAP	Ground Truth
Water is running from a faucet into a sink	A person is walking through a pile of leaves	Someone is walking outside on a path covered with dried leaves
The wind is blowing and a car is driving by	A group of children are yelling and screaming	Many children are talking and screaming, all at the same time
A heavy rain coming down outside during a storm	A saw is being used to cut a piece of wood	A saw being used to saw wood that makes squeaking noises at the end
	Generation Scheme	
Beam search	Reranking	Ground Truth
A gun is being fired at a target	A hammer is repeatedly hit with a metal object	Someone is repeatedly hitting a hammer onto a wall or a nail
Birds are chirping and people are talking in the background	Children are playing, a car is driving, and birds are chirping	Children shout and play at the playground as cars loudly drive by in the background
The engine of a car starts and then the car drives away	A motorcycle engine starts up and idles for a while before idling down and idling again	A motorcycle engine starts and idles for a while

Table 3: Result on AudioCaps

Model	METEOR	CIDEr	SPICE	SPIDEr	FENSE
AL-MixGen [29]	0.242	0.769	0.181	0.475	-
Wavcaps [17]	0.250	0.787	0.182	0.485	-
CoNeTTE [5]	0.253	0.806	0.184	0.495	0.643
EnCLAP-base [14]	0.247	0.780	0.186	0.483	0.650
EnCLAP-large [14]	0.255	0.803	0.188	0.495	0.655
EnCLAP++-base	0.257	0.815	0.188	0.501	0.661
EnCLAP++-large	**0.269**	**0.823**	**0.197**	**0.510**	**0.665**

of details. The inclusion of DAC, a timestep-level representation, enables the model to capture more fine-grained details of the scene.

Sequence-level Acoustic Embedding. While the model without CLAP generally succeeds in capturing the atmosphere of the acoustic scene, it tends to confound the overall semantic meaning of the scene. Thus, its captions describe an event similar to the actual event, but is actually different. We believe this comes from the lack of world knowledge to clear up the ambiguity. Thus, the variant with CLAP does not suffer from this issue. We attribute this to the model's lack of world knowledge, which fails to resolve ambiguities. Consequently, its generated captions describe an event that is similar to, yet distinct from, the actual event. In contrast, the variant with CLAP does not suffer from this issue.

Generation and Reranking. The captions produced by beam search variants are typically shorter and more concise, often omitting scene details. In contrast, the reranking variant generates more detailed captions that closely align with the label captions.

3.6. Results on AudioCaps

Based on observations from Section 3, we propose EnCLAP++, an improved version of EnCLAP that incorporates DAC, large-scale pretraining, and hybrid reranking. We evaluate EnCLAP++ on the AudioCaps dataset and present the results in Table 3. The assessment shows that both EnCLAP++-base and EnCLAP++-large outperform their respective EnCLAP counterparts, demonstrating the effectiveness of our mix of optimizations across different datasets.

Table 4: DCASE 2024 Challenge Result on Clotho Evaluation Split

Model	METEOR	CIDEr	SPICE	SPIDEr	FENSE
DCASE 2024 Baseline	0.186	0.442	0.135	0.288	0.510
Feng *et al.* [30]	0.192	0.495	0.141	0.318	0.525
Kim *et al.* [31]	0.189	0.409	0.135	0.272	0.526
Liu *et al.* [32]	0.195	0.493	0.145	0.319	0.533
Chen *et al.* [33]	0.194	**0.509**	0.145	**0.327**	0.541
Jung *et al.* [34]	0.172	0.344	0.140	0.242	**0.554**
EnCLAP++	**0.199**	0.480	**0.148**	0.314	0.544

3.7. Results on DCASE Challenge 2024

We submitted a variant of EnCLAP++ to the DCASE Challenge 2024. This variant employs a large version of BART and is pretrained on an extensive dataset that combines WavCaps, AudioCaps, and Clotho-Chatmix [6]. Due to the challenge regulations, we could not use CLAP because of potential overlap with the evaluation dataset. Therefore, we adopted CNext from Sec 2.2, which was additionally trained with text-retrieval on WavCaps, AudioCaps, and Clotho, as the sequence-level representation.

The overall results are presented in Table 4. Our model achieved second place in the challenge, which was ranked based on the FENSE metric. Additionally, our model outperformed all other models on the METEOR and SPICE metrics.

4. CONCLUSION

This study presents an analysis of the EnCLAP framework and its components. Our investigation reveals that replacing the EnCodec encoder with the DAC encoder, augmenting the pretraining dataset with large-scale weakly-labeled data, and the incorporating of a reranking scheme enhances the model's performance in audio captioning. Notably, our modified variant, EnCLAP++ shows significant improvement over the original model. Future directions for our research involve extending the EnCLAP framework to incorporate recent advances in large language models, thereby enhancing its capabilities.

5. REFERENCES

[1] K. Drossos, S. Adavanne, and T. Virtanen, "Automated audio captioning with recurrent neural networks," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2017.

[2] C. D. Kim, B. Kim, H. Lee, and G. Kim, "AudioCaps: Generating captions for audios in the wild," in *NAACL*, 2019.

[3] K. Drossos, S. Lipping, and T. Virtanen, "Clotho: an audio captioning dataset," in *ICASSP*, 2020.

[4] X. M. et al., "Audio captioning transformer," in *DCASE Workshop*, 2021.

[5] E. Labbé, T. Pellegrini, and J. Pinquier, "Conette: An efficient audio captioning system leveraging multiple datasets with task embedding," *arXiv preprint arXiv:2309.00454*, 2023.

[6] S.-L. Wu, X. Chang, G. Wichern, J.-W. Jung, F. Germain, J. Le Roux, and S. Watanabe, "Improving audio captioning models with fine-grained audio features, text embedding supervision, and llm mix-up augmentation," in *ICASSP*, 2024.

[7] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, 2019.

[8] M. Kim, S.-B. Kim, and T.-H. Oh, "Prefix tuning for automated audio captioning," in *ICASSP*, 2023.

[9] S. Deshmukh, B. Elizalde, R. Singh, and H. Wang, "Pengi: An audio language model for audio tasks," *arXiv:2305.11834*, 2023.

[10] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *ACL*, 2020.

[11] F. Gontier, R. Serizel, and C. Cerisara, "Automated audio captioning by fine-tuning bart with audioset tags," in *DCASE Workshop*, 2021.

[12] Y. Koizumi, R. Masumura, K. Nishida, M. Yasuda, and S. Saito, "A Transformer-Based Audio Captioning Model with Keyword Estimation," in *INTERSPEECH*, 2020.

[13] E. Labbé, J. Pinquier, and T. Pellegrini, "Multitask learning in audio captioning: a sentence embedding regression loss acts as a regularizer," *arXiv preprint arXiv:2305.01482*, 2023.

[14] J. Kim, J. Jung, J. Lee, and S. H. Woo, "Enclap: Combining neural audio codec and audio-text joint embedding for automated audio captioning," in *ICASSP*, 2024.

[15] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," *arXiv:2210.13438*, 2022.

[16] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, "Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation," in *ICASSP*, 2023.

[17] X. Mei, C. Meng, H. Liu, Q. Kong, T. Ko, C. Zhao, M. D. Plumbley, Y. Zou, and W. Wang, "Wavcaps: A chatgpt-assisted weakly-labelled audio captioning dataset for audio-language multimodal research," *arXiv:2303.17395*, 2023.

[18] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, "Soundstream: An end-to-end neural audio codec," *IEEE/ACM TASLP*, 2021.

[19] R. Kumar, P. Seetharaman, A. Luebs, I. Kumar, and K. Kumar, "High-fidelity audio compression with improved rvqgan," in *NeurIPS*, 2023.

[20] F. Kreuk, G. Synnaeve, A. Polyak, U. Singer, A. Défossez, J. Copet, D. Parikh, Y. Taigman, and Y. Adi, "Audiogen: Textually guided audio generation," in *ICLR*, 2022.

[21] C. Wang, S. Chen, Y. Wu, Z. Zhang, L. Zhou, S. Liu, Z. Chen, Y. Liu, H. Wang, J. Li, *et al.*, "Neural codec language models are zero-shot text to speech synthesizers," *arXiv:2301.02111*, 2023.

[22] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Defossez, "Simple and controllable music generation," in *NeurIPS*, 2023.

[23] H. Wu, H.-L. Chung, Y.-C. Lin, Y.-K. Wu, X. Chen, Y.-C. Pai, H.-H. Wang, K.-W. Chang, A. H. Liu, and H.-Y. Lee, "Codec-superb: An in-depth analysis of sound codec models," *arXiv preprint arXiv:2402.13071*, 2024.

[24] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *ICASSP*, 2017.

[25] T. Pellegrini, I. Khalfaoui-Hassani, E. Labbé, and T. Masquelier, "Adapting a convnext model to audio classification on audioset," *arXiv:2306.00830*, 2023.

[26] M. Luong, K. Nguyen, N. Ho, R. Haf, D. Phung, and L. Qu, "Revisiting deep audio-text retrieval through the lens of transportation," in *ICLR*, 2024.

[27] S. Xiao, Z. Liu, P. Zhang, and N. Muennighof, "C-pack: Packaged resources to advance general chinese embedding," *arXiv preprint arXiv:2309.07597*, 2023.

[28] Z. Zhou, Z. Zhang, X. Xu, Z. Xie, M. Wu, and K. Q. Zhu, "Can audio captions be evaluated with image caption metrics?" in *ICASSP*, 2022.

[29] E. Kim, J. Kim, Y. Oh, K. Kim, M. Park, J. Sim, J. Lee, and K. Lee, "Exploring train and test-time augmentations for audio-language learning," *arXiv:2210.17143*, 2022.

[30] Q. Feng, Q.and Kong, "Semantic enhancement encoder for audio captioning and spectrogram-based data augmentation," DCASE Challenge, Tech. Rep., 2024.

[31] E. Kim, J. Sim, J. W. Lee, and K. Lee, "Retrieval-augmented audio captioning with llm fine-tuning," DCASE Challenge, Tech. Rep., 2024.

[32] J. Liu and G. Li, "Leveraging ced encoder and large language models for automated audio captioning," DCASE Challenge, Tech. Rep., 2024.

[33] W. Chen, X. Li, Z. Ma, Y. Liang, A. Jiang, Z. Zheng, Y. Qian, P. Fan, W.-Q. Zhang, C. Lu, J. Liu, and X. Chen, "Sjtu-thu automated audio captioning system for dcase 2024," DCASE Challenge, Tech. Rep., 2024.

[34] J.-w. Jung, D. Zhang, H. C.-H. Yang, S.-L. Wu, D. M. Chan, Z. Kong, D. Ruifan, Z. Yaqian, V. Rafael, and S. Watanabe, "Automatic audio captioning with encoder fusion, multi-layer aggregation, and large language model enriched summarization," DCASE Challenge, Tech. Rep., 2024.

CLAP4SED: TRAINING-FREE MULTIMODAL FEW-SHOT RETRIEVAL FOR REAL-TIME SOUND EVENT DETECTION ON EMBEDDED DEVICES

Wei-Cheng Lin, Irtsam Ghazi, Ajit Belsarkar, Luca Bondi, Samarjit Das, Ho-Hsiang Wu

Robert Bosch LLC, USA
wei-cheng.lin@us.bosch.com

ABSTRACT

Implementing real-time *sound event detection* (SED) on embedded devices poses significant challenges, primarily related to generalizability and complexity. Existing SED models are predominantly suited for closed-form recognition, making adaptation to new or unseen sound classes difficult. While recent advancements in *audio foundation models* (AFM) such as CLAP offer potential for open-form sound event classification, they often come with substantial model complexity, rendering them impractical to deploy on embedded devices for real-time tracking. In this study, we introduce the CLAP4SED framework, a training-free, real-time SED solution derived from CLAP that can be flexibly deployed across various open-ended scenarios on embedded devices. Our experimental results conducted on three publicly available datasets demonstrating the competitive SED accuracy with less than 100ms latency under Ambarella CV22 camera chip setup.

Index Terms— sound event detection, few-shot prompt engineering, audio foundation models, embedded AI systems

1. INTRODUCTION

Audio has become a popular sensory modality for monitoring our environment, it complements vision in better handling of occlusions and can support omnidirectional signal. Audio sensors have been deployed to real-world environment for applications such as noise monitoring in urban areas [1], tracking avian diversities [2] and bird migrations [3]. These *sound event detection* (SED) [4] solutions are usually deployed as embedded systems with computational resource constraints, requiring constant monitoring and handling of input signal streams, and supporting diverse characteristics of sounds such as gunshot [5], glass breaking, baby crying, and screaming [6], etc.

Recent advancements in *audio foundation models* (AFM) provide a promising solution to bridge the generalization gaps encountered with unseen acoustic events or conditions. There are two main campaigns for building AMF: First, *contrastive language-audio pretraining* (CLAP) [7], trained with large amount of audio captioning data [8, 9] contrastively, sometimes with the aid of ChatGPT-assisted caption generation [10]. Second, audio encoders are trained to adapt towards *large language models* (LLMs) such as Pengi [11], *listen, think, and understand* (LTU) [12], Qwen-Audio [13], and SALMONN [14]. These AFM unlock free-form natural language interactions with audio data and provide new avenues for embedded audio AI solutions. There has also been a paradigm shift from collecting data tailored for specific downstream tasks and training models in a supervised manner to utilizing these AFM for rapid prototyping with zero-shot capabilities, and further adapting with few-shot examples [15, 16]. However, AFM typically rely on computational heavy model architectures, especially when they accompany with additional language models. This imposes another critical challenge for utilizing AFM under the embedded device setups [17].

Recently, it has been a surge of interest in adapting the CLAP model for offline audio analytic techniques, such as zero-shot audio classification or retrieval via natural language prompts [18, 19]. However, there is a noticeable gap in utilizing CLAP for on-device real-time applications such as SED. To bridge this gap, we propose CLAP4SED in this study, which is a method that utilizes a pretrained lightweight CLAP model for real-time SED tasks. This approach is designed to be executable on embedded devices, facilitating flexible adaptation of SED to handle various deployment environments. More specifically, we decouple the query step from the original CLAP inference stage and devise an offline multimodal few-shot retrieval pipeline to achieve real-time SED. We experiment with several prompting strategies from zero-shot to few-shot scenarios and discuss corresponding constraints in practical applications. We also highlight several design choices and trade-offs deploying these SED models to real-world embedded devices. The main contributions of this study are:

- We propose a training-free, real-time SED solution based on the novel multimodal retrieval framework, which aims to be executable on embedded devices for practical deployment.
- We provide comprehensive experimental results and highlight the design choices between the model performance and complexity for CLAP4SED.
- To best of our knowledge, we are the first work that explicitly leverages CLAP to perform on-device real-time SED.

2. PROPOSED FRAMEWORK

The proposed full framework consists of two main steps: A). building a backbone AFM optimized for operation on embedded devices, B). the multimodal few-shot retrieval system to perform real-time SED predictions.

2.1. CLAP Pretraining

We implement the CLAP model as our AFM for the first step. The CLAP training involves in audio $f_A(\cdot)$ and text $f_T(\cdot)$ encoders to process incoming pairs of audio sequence X_a and the corresponding caption descriptions X_t. This results in the audio $E_a = f_A(X_a)$ and text $E_t = f_T(X_t)$ embeddings, respectively. The model is then trained to optimize the symmetric similarity contrastively (Eq. 1) in a joint multimodal space for audio-text pairs containing within a mini batch size B, where η is a temperature parameter to scale the output ranges. More details can be found in [7].

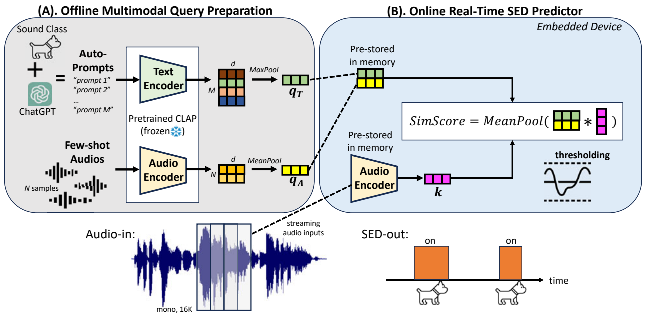


Figure 1: Overview of the proposed CLAP4SED framework for real-time SED on embedded device.

$$\mathcal{L} = \frac{1}{2B}\sum_{B}[\log diag(softmax(\eta(E_a \cdot E_t^{\top}))) + \log diag(softmax(\eta(E_t \cdot E_a^{\top})))] \quad (1)$$

However, existing pretrained CLAP models [10, 20] are majorly focusing on recognition performance without much consideration of model complexity, which might not be affordable to deploy on embedded devices. To accommodate this restriction, we substitute the conventional audio encoder from Transformers-based (e.g., HTSAT [21]) to lightweight CNN-based (PANNs [22]) family architecture. Since compare or compete embedded *machine learning* (ML) approaches is not our paper focus, we choose a relatively naïve method to obtain lightweight encoder for simplicity of the proposed framework. While beyond the scope of this study, it's worth noting that various advanced model compression techniques such as quantization or distillation [23] could be considered to further improve the backbone AFM performance. As it is unavoidable to balance model efficiency with performance, we present Table 1 to benchmark our retrained lightweight CLAP encoder, providing a reference for this trade-offs. For other training configurations, we follow closely the standard recipe of CLAP works [7, 20] and discuss in Section 3.1.

2.2. CLAP4SED: Multimodal Few-shot Retrieval for SED

The core idea to leverage a retrieval-based AFM for SED is to decouple the query component from the original CLAP inference stage. Figure 1 provides an overview of this framework. Specifically, the desired queries are calculated offline and stored in advance on the embedded devices. This approach eliminates the model complexity of the entire text modality (i.e., LLMs), thereby substantially lowering memory and computational demands and enabling operation on small embedded devices. However, the robustness and representativeness of pre-computed queries emerge as the most crucial factors for accurate SED predictions.

A). Offline Query Preparation: we propose to utilize multimodal information for obtaining effective query prototypes (see the top-left gray box in Figure 1), assuming N few-shot audio samples are available on hand per interested sound event. For the audio part, the trained audio encoder $f_A(\cdot)$ is applied to extract audio embeddings from the given few-shot samples. Following by a mean-pooling operation to summarize the audio prototypical vector as the final audio query $q_A \in \mathbb{R}^{1\times d}$, where d represents the dimension of hidden space that performs audio-text contrastive learning in the CLAP model. As for the text query preparation, we first employ GPT-4[1] to rewrite the convention CLAP retrieval template (e.g., "*this is the sound of [class label]*") for enriching text expressions [10] into M different prompts. These gpt-generated retrieval prompts are then fed into the trained text encoder $f_T(\cdot)$ to obtain embeddings. An audio-informed max-pooling operation over prompts is conducted, which only returns the most relevant (i.e., maximum dot-product similarity) prompt embedding to the given few-shot audio embeddings. This results in the final text query $q_T \in \mathbb{R}^{1\times d}$. From the high-level standpoint, audio query guides the specificity of retrieval outcomes while text query enhances additional diversity from different modality perspective for better robustness.

B). Online SED Predictor: only the lightweight audio encoder (Sec. 2.1) and modality-specific query vectors need to be pre-stored in the embedded device, as depicted in the top-right blue box in Figure 1. Upon receiving an input audio streaming data chunk (with window size L), encoder $f_A(\cdot)$ extracts it to generate key embeddings $k \in \mathbb{R}^{1\times d}$, which is then used to calculate a predefined similarity criteria with the prepared queries, thereby forming the averaged decision score across modalities. Finally, a simple binary thresholding is applied to determine the activity of sound event for that specific timeframe. The minimum real-time prediction time grid, denoted as τ, depends on the overall latency of the prediction process.

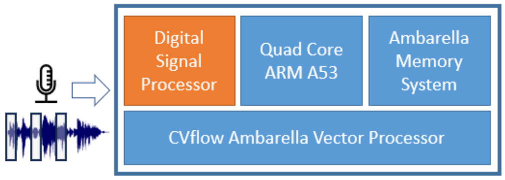


Figure 2: High-level structure of the embedded hardware setup based on Ambarella CV22 chip.

3. EXPERIMENTAL SETUPS

3.1. Embedded System, Pretraining and Configurations

We use the Ambarella CV22 chip [24] to construct the embedded system environment, which is typically used for IP cameras. The CV22 chip comes equipped with a quad core ARM A-53 Linux enabled processor, 1MB L2 cache, an Neon SIMD accelerator for *digital signal processing* (DSP), and a *computer vision* (CV) flow vector processor for deep learning matrix operations. The Neon chip can effectively accelerate the *Fast-Fourier transform* (FFT) for spectrogram computations. Figure 2 shows a high-level structure of the hardware components we used for running computational cost analysis in Section 4.3.

For the CLAP model pretraining, we use Adam (lr=0.0001) to optimize the standard contrastive loss (Eq. 1) based on the AudioCaps, Clotho, FSD50K, MACS, and WavCaps [10] train datasets. The default audio encoder $f_A(\cdot)$ is PANN10 [22] architecture unless specified in the results. We use the pretrained CLIP [25] text encoder to extract caption embeddings, the encoder $f_T(\cdot)$ is frozen all

[1] https://openai.com/gpt-4

the time during the training process. The hidden dimension d of the joint contrastive space is 512, temperature η is 0.07, and 128 batch size B training on a single NVIDIA-TESLA-V100 32GB GPU device. All the models are implemented in PyTorch.

For the real-time SED configurations, the streaming audio input is a 1 sec length (i.e., the sliding window size L), 16K, mono and 16-bit data chunk. Note that longer lengths require to register more memory buffer and increase the computational latency. The time grid of producing SED outputs is set to 0.1 secs (i.e., the window hop size τ), since our maximum prediction latency can be less than 100ms under the proposed framework. We assume 5-shot examples (N) are available per sound event in default unless specified. These few-shot samples are randomly selected from the corresponding validation or train data. We prompt GPT-4 with "*what are the sounds of [class label] ?*" to produce 30 (M) diversified but relevant enough sound descriptions for each target retrieval class. Cosine similarity is set as the criteria to measure the decision score.

Since our proposed framework is to perform real-time SED under practical industrial setup (e.g., security camera), the model is not receiving the full clip (global)-level context to compute advanced offline metrics such as PSDS scores [26]. Instead, the model only receives local segment input (e.g., 1 sec streaming chunk) during each inference period. Therefore, we calculate the *area under curve* (AUC) for segment-based precision-recall as our system evaluation metric, which is more suitable to evaluate on-device real-time SED and can comprehensively compare overall performance for the full threshold space.

3.2. Datasets

One important feature of the proposed framework is that we can quickly adapt the trained AFM towards different application scenarios without involving additional finetuning or retraining efforts on the embedded devices (i.e., training-free approach). Here, we showcase this flexibility by evaluating it on three diverse datasets for the domestic environments, urban sounds, and aggression events monitoring, respectively.

- **DESED** [27]: is composed of 10 domestic event classes (i.e., alarm/bell/ringing, blender, cat, dog, dishes, electric shaver/toothbrush, frying, running water, speech and vacuum cleaner) originating from the AudioSet [28]. We only utilize its evaluation set to report the system performance, which has 692 audio files (fixed 10 secs length for each) in total.
- **Urban-SED** [29]: synthesizes strong-labeled soundscapes from UrbanSound8K dataset using the SCAPER [29] tool, which consists of 10 city sounds (i.e., air conditioner, car horn, children playing, dog bark, drilling, engine idling, gunshot, jackhammer, siren and street music). Its evaluation set contains 2,000 audio files and each is 10 secs long.
- **Aggression-SED**: is our own curated evaluation subset by defining the violent or aggressive relevant events out of the AudioSet. We define 8 classes to be included (i.e., smoke/fire/car alarm, ambulance/defense/truck siren, explosion, fire, gunshot, screaming, shouting and smash/crash/breaking sound), which has a total of 1,495 audio files extracting from the AudioSet strong evaluation partition[2].

Table 1: Performance summary of our retrained lightweight CLAP encoders comparing to existing SOTA models. We evaluate the zero-shot classification (ZS) on UrbanSound8k (US8K) and ESC-50 based on macro F1 score (F1), as well as the text-to-audio (T2A) and audio-to-text (A2T) retrieval on Clotho using recall at 10 (R@10) metrics. All the results are in percentage scale.

	US8K (F1)	**ECS-50** (F1)	**Clotho** (R@10)	
	ZS	*ZS*	*T2A*	*A2T*
PANN14 [7]	73.2	82.6	-	-
HTSAT-LAION [20]	77.0	91.0	54.4	65.7
HTSAT-WavCaps [10]	80.6	94.8	50.9	56.6
PANN6 (ours)	68.1	68.9	33.8	35.1
PANN10 (ours)	72.5	78.0	37.8	42.3
PANN14 (ours)	77.7	85.3	42.5	48.1

3.3. Ablation Baselines

We want to highlight that our approach is incomparable to existing SED models, since we do not rely on any labeled data (except for a very limited few-shot audio examples) nor a particular training framework for SED. Instead, we conduct comparisons against ablation baselines focusing on the query design component to demonstrate the advantage of leveraging multimodal information for retrieval-based SED. Specifically, four single-modality baselines are compared by preparing the query vector q_T or q_A in different ways while everything else remains the same. These single-modality retrieval approaches are also commonly adopted from previous literatures.

- **Class Prompt**: zero-shot audio retrieval using raw class labels (i.e., "*[class label]*") as input prompt for generating the text-only query q_T, denoted as *text-class*.
- **Template Prompt** [7]: zero-shot audio retrieval appending with natural language-alike template (i.e., "*this is the sound of [class label]*") to produce the text-only query q_T, denoted as *text-temp*.
- **GPT Prompt** [10]: same as we depicted in Figure 1 (the gray box) but only considers the text-only query q_T. We use mean-pooling operation instead of audio-informed max-pooling to summarize the query embedding, since we do not have available audio samples under the single-modality setting to compute the most relevant prompt. We denote this baseline as *text-gpt*.
- **Audio Prototypes** [30]: same as we depicted in Figure 1 (the gray box) but only considers the audio-only query q_A to form an audio-to-audio retrieval task. As q_A serves as the prototypical vector of audios, we denote it as *audio-proto*.

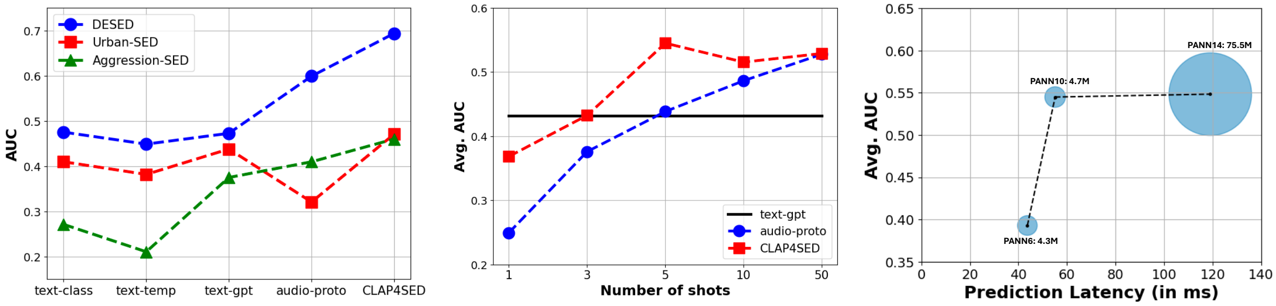
4. EXPERIMENTAL RESULTS AND ANALYSIS

4.1. System Performance Comparison

Figure 3(a) summarizes the evaluation results of single-modality baselines (Sec. 3.3) versus proposed CLAP4SED method across three datasets (Sec. 3.2). There are three major points to focus on:

First, we can observe that the text-gpt approach generally obtains higher performance comparing to other text-only query methods (i.e., text-class and text-temp), especially for the Aggression-SED. It might due to LLMs can effectively enrich language diver-

[2] https://research.google.com/audioset/download_strong.html



(a) (b) (c)

Figure 3: Comparison of system performance under different scenarios and model configurations. (a) Performance across SED scenarios for different queries. (b) Influence of few-shot numbers. (c) Performance and complexity trade-offs.

sity to incorporate a broader spectrum of common sense knowledge into the query space. This augmentation contributes to a more comprehensive coverage of sound event scenarios, increasing the model's generalizability.

Second, we can see that the audio-proto approach can outperform text-only schemes in the cases of DESED and Aggression-SED. However, its performance significantly deteriorates in the Urban-SED scenario, indicating a notable issue with robustness. While few-shot references can offer advantages for specific prototypes (i.e., retrieval specifications), they also impose limitations on generalization capability as these collected few samples on hand might not be sufficient to represent the full event space. This compromises the model's robustness against diverse scenarios, limiting its practical applicability.

Last, the proposed CLAP4SED framework effectively reconciles the trade-offs inherent in both text-only and audio-only approaches by leveraging the benefits of multimodal fusion. The text-gpt query q_T enhances model generalization, addressing potential representational gaps in the audio-proto q_A. Meanwhile, the audio-proto offers supplementary guidance on recognition precision, thereby complementing each other's information. As a result, CLAP4SED consistently achieves the best system performance across three datasets over all the single-modality approaches.

4.2. Few-shot Capability Analysis

This section discusses how the few-shot number (N) impacts on system performance that involves in utilizing audio samples as query (i.e., audio-proto and CLAP4SED). Figure 3(b) illustrates the averaged AUC results across three datasets, and we pick text-gpt as the benchmark representative since it obtains the most competitive performance among the text-only query approaches. We can observe a consistent trend where increasing the number of few-shot audios leads to an improvement in overall SED performance. Upon gathering 5-shot examples, both audio-proto and CLAP4SED outperform the text-only query approaches. Interestingly, the proposed CLAP4SED demands significantly fewer audio samples compared to the audio-proto approach. Its performance with 5-shots can achieve comparable results to 50-shots for audio-proto (this trend holds true for 1-shot and 3-shots cases as well). This characteristic has significant importance for practical applications, as it is often infeasible to gather as many supervised shots in most cases. With the advantage to collect just 5 examples for new environments or undefined sound events, CLAP4SED can rapidly deploy and adapt to various real-world scenarios by simply updating the query vectors without additional training steps, resulting in an effective training-free solution.

4.3. Computational Trade-Offs

We also provide the computational trade-offs of CLAP4SED based on the configured embedded system setup (Sec 3.1) as a reference for future development. Figure 3(c) visualizes the result for PANN6, PANN10 and PANN14 architectures. The radius of a circle indicates the model size in number of parameters, x-axis represents overall prediction latency in milliseconds and y-axis shows the corresponding SED performance in averaged AUC. We can see that there is a clear sweet point of using the PANN10 encoder. It significantly improves the overall recognition accuracy from PANN6 with acceptable model size (4.3M to 4.7M) and latency (45ms to 55ms) increases. On the other hand, PANN14 only brings a very limited performance improvement from PANN10, but drastically escalates the computational requirements (e.g., latency increases from 55ms to 120ms). Prediction latency is a critical factor that cannot be compromised in real-time detection setups. Therefore, PANN10 emerges as the most recommended encoder for the CLAP4SED framework, striking a balance between recognition performance and computational efficiency.

5. CONCLUSION

In this study, we introduce the CLAP4SED framework, which utilizes the CLAP foundation model to enable real-time SED on embedded devices. The core innovation lies in decoupling the query component from the CLAP retrieval pipeline. This allows for significant reduction in model complexity and flexible adaptation to various SED scenarios, resulting in an efficient training-free solution. Notably, our experimental results showcase the effectiveness of the proposed few-shot multimodal query approach, which effectively combines the advantages of both text and audio modalities, thereby bridging modality gaps. Additionally, we provide comprehensive design choices and trade-offs analysis as a reference for future development endeavors.

6. REFERENCES

[1] J. P. Bello, C. Silva, O. Nov, R. L. Dubois, A. Arora, J. Salamon, C. Mydlarz, and H. Doraiswamy, "SONYC: A system for monitoring, analyzing, and mitigating urban noise pollution," *Communications of the ACM*, vol. 62, no. 2, pp. 68–77, 2019.

[2] S. Kahl, C. M. Wood, M. Eibl, and H. Klinck, "BirdNET: A deep learning solution for avian diversity monitoring," *Ecological Informatics*, vol. 61, p. 101236, 2021.

[3] V. Lostanlen, A. Cramer, J. Salamon, A. Farnsworth, B. M. Van Doren, S. Kelling, and J. P. Bello, "BirdVox: Machine listening for bird migration monitoring," *bioRxiv*, pp. 2022–05, 2022.

[4] A. Mesaros, T. Heittola, T. Virtanen, and M. D. Plumbley, "Sound event detection: A tutorial," *IEEE Signal Processing Magazine*, vol. 38, no. 5, pp. 67–83, 2021.

[5] D. Mares and E. Blackburn, "Acoustic gunshot detection systems: a quasi-experimental evaluation in st. louis, mo," *Journal of experimental criminology*, vol. 17, pp. 193–215, 2021.

[6] A. Suliman, B. Omarov, and Z. Dosbayev, "Detection of impulsive sounds in stream of audio signals," in *2020 8th International Conference on Information Technology and Multimedia (ICIMU)*. IEEE, 2020, pp. 283–287.

[7] B. Elizalde, S. Deshmukh, M. Al Ismail, and H. Wang, "CLAP: Learning audio concepts from natural language supervision," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[8] C. D. Kim, B. Kim, H. Lee, and G. Kim, "AudioCaps: Generating captions for audios in the wild," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 119–132.

[9] K. Drossos, S. Lipping, and T. Virtanen, "Clotho: An audio captioning dataset," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 736–740.

[10] X. Mei, C. Meng, H. Liu, Q. Kong, T. Ko, C. Zhao, M. D. Plumbley, Y. Zou, and W. Wang, "WavCaps: A chatgpt-assisted weakly-labelled audio captioning dataset for audio-language multimodal research," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 3339–3354, 2024.

[11] S. Deshmukh, B. Elizalde, R. Singh, and H. Wang, "Pengi: An audio language model for audio tasks," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[12] Y. Gong, H. Luo, A. H. Liu, L. Karlinsky, and J. R. Glass, "Listen, think, and understand," in *The Twelfth International Conference on Learning Representations*, 2024.

[13] Y. Chu, J. Xu, X. Zhou, Q. Yang, S. Zhang, Z. Yan, C. Zhou, and J. Zhou, "Qwen-Audio: Advancing universal audio understanding via unified large-scale audio-language models," *arXiv preprint arXiv:2311.07919*, 2023.

[14] C. Tang, W. Yu, G. Sun, X. Chen, T. Tan, W. Li, L. Lu, Z. MA, and C. Zhang, "SALMONN: Towards generic hearing abilities for large language models," in *The Twelfth International Conference on Learning Representations*, 2024.

[15] Z. Lin, S. Yu, Z. Kuang, D. Pathak, and D. Ramanan, "Multimodality helps unimodality: Cross-modal few-shot learning with multimodal models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 19 325–19 337.

[16] S. Huang, L. Dong, W. Wang, Y. Hao, S. Singhal, S. Ma, T. Lv, L. Cui, O. K. Mohammed, B. Patra, *et al.*, "Language is not all you need: Aligning perception with language models," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[17] Y. Chen, B. Zheng, Z. Zhang, Q. Wang, C. Shen, and Q. Zhang, "Deep learning on mobile and embedded devices: State-of-the-art, challenges, and future directions," *ACM Computing Surveys (CSUR)*, vol. 53, no. 4, pp. 1–37, 2020.

[18] J. Liang, X. Liu, H. Liu, H. Phan, E. Benetos, M. D. Plumbley, and W. Wang, "Adapting language-audio models as few-shot audio learners," in *Proc. INTERSPEECH 2023*, 2023, pp. 276–280.

[19] W.-C. Lin, S. Ghaffarzadegan, L. Bondi, A. Kumar, S. Das, and H.-H. Wu, "CLAP4Emo: Chatgpt-assisted speech emotion retrieval with natural language supervision," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2024)*, Seoul, Korea, April 2024, pp. 11 791–11 795.

[20] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, "Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[21] K. Chen, X. Du, B. Zhu, Z. Ma, T. Berg-Kirkpatrick, and S. Dubnov, "HTS-AT: A hierarchical token-semantic audio transformer for sound classification and detection," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 646–650.

[22] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "PANNs: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2880–2894, 2020.

[23] A. Polino, R. Pascanu, and D. Alistarh, "Model compression via distillation and quantization," in *International Conference on Learning Representations*, 2018.

[24] Ambarella Inc., "Product brief on ambarella cv22: Computer vision soc for ip cameras," 2021.

[25] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *Proceedings of the 38th International Conference on Machine Learning*, vol. 139. PMLR, 18–24 Jul 2021, pp. 8748–8763.

[26] Ç. Bilen, G. Ferroni, F. Tuveri, J. Azcarreta, and S. Krstulović, "A framework for the robust evaluation of sound event detection," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 61–65.

[27] N. Turpault, R. Serizel, A. P. Shah, and J. Salamon, "Sound event detection in domestic environments with weakly labeled data and soundscape synthesis," in *Workshop on Detection and Classification of Acoustic Scenes and Events*, 2019.

[28] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio Set: An ontology and human-labeled dataset for audio events," in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 776–780.

[29] J. Salamon, D. MacConnell, M. Cartwright, P. Li, and J. P. Bello, "Scaper: A library for soundscape synthesis and augmentation," in *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2017, pp. 344–348.

[30] S. Deshmukh, B. Elizalde, and H. Wang, "Audio retrieval with WavText5K and CLAP training," in *Proc. INTERSPEECH 2023*, 2023, pp. 2948–2952.

GUIDED CAPTIONING OF AUDIO

Irene Martín-Morató, James Afolaranmi, Annamaria Mesaros

Signal Processing Research Centre, Tampere University, Finland
{irene.martinmorato@tuni.fi, james.afolaranmi@tuni.fi, annamaria.mesaros@tuni.fi

ABSTRACT

This work introduces a guided captioning system that aims to produce captions focused on different audio content, depending on a guiding text. We show that using keywords guidance results in more diverse captions, even though the usual captioning metrics do not reflect this. We design a system that can be trained using keywords automatically extracted from reference annotations, and which is provided with one keyword at test time. When trained with 5 keywords, the produced captions contain the exact guidance keyword 70% of the time, and results in over 3600 unique sentences for Clotho dataset. In contrast, a baseline without any keywords produces 700 unique captions on the same test set.

Index Terms**—** automatic audio captioning

1. INTRODUCTION

Automatic audio captioning (AAC) is a cross-modal task combining audio signal analysis and natural language processing [1]. Captioning differs from other audio analysis tasks such as detection or classification because it requires not only identification of the sounds, but also a description of the relationships between co-occurring events. Textual descriptions provide more information about the audio content than simple labels, indicating for example which sounds are more prominent and which ones are background, how sounds co-occur or follow each other, or describe attributes, e.g. how loud/quiet or far/near the sound is.

What defines a good caption is subject to the specific situation. Generally speaking, sensory descriptions have as primary function transmitting the main information, which for audio captioning is likely be the main sound event; but the way this information is included in a caption is very subjective [2]. AAC datasets provide captions for training the systems, one or multiple captions per clip [3–5], reflecting to some extent the fact that different descriptions of the same audio clip are correct, even though not identical.

AAC systems are trained in a supervised manner, being fed with the audio file and its corresponding reference captions [6, 7]; evaluation is performed by comparing an automatically predicted caption against the reference captions, to measure how well the predicted caption matches each of the reference captions. Researchers have questioned the use of machine translation or image captioning metrics for evaluating audio captions, because the auditory, temporal and spatial properties of the sound are not the same as objects' properties. As a result, multiple captioning metrics were proposed specifically for AAC, e.g. FENSE [8], SPICE+ [9], CB-score [10], SPIDEr-max [11]. However, the status quo in AAC is still dominated by small training datasets, limited vocabulary, and unclear interpretation of the metrics.

The concept of "guiding text" for captioning has been investigated in [12]; the authors proposed "conceptual captions", where a provided text controls what an image captioning system should focus on. A similar approach was used in [13] for AAC; the authors used a transformer with keyword estimation to generate a caption that contains the estimated keyword. In [14], the authors used keywords estimated from the given audio clip through automatic audio tagging. Furthermore, Xu et.al. [15] focus on improving diversity of the captions without decreasing accuracy. These works focused on improving AAC performance as evaluated with the usual AAC metrics. However, captions containing words from the reference vocabulary will usually have high scores, even if they do not completely describe the audio content; moreover, high semantic similarity does not necessarily reflect the true correspondence with the described sounds, as observed in [10].

The contributions of this paper are as follows: (1) a guided captioning system that can be trained with an arbitrary list of keywords per audio clip; (2) a systematic study of the effect of keywords on the predicted captions. Rather than improving AAC performance in terms of the usual metrics, as done in previous works, we focus on guiding the system towards a specific sound event of interest: a user interested in one particular event will provide a keyword as guidance in order to obtain a description of the specific event. This description may be only partial as to the acoustic content of the clip, but correct and desired by the user. Our experiments show that given the same audio clip as input, it is possible to

This work was supported by Jane and Aatos Erkko Foundation under grant number 200061, "Guided captioning for complex acoustic environments". The authors wish to thank CSC-IT Centre of Science Ltd., Finland, for providing computational resources.

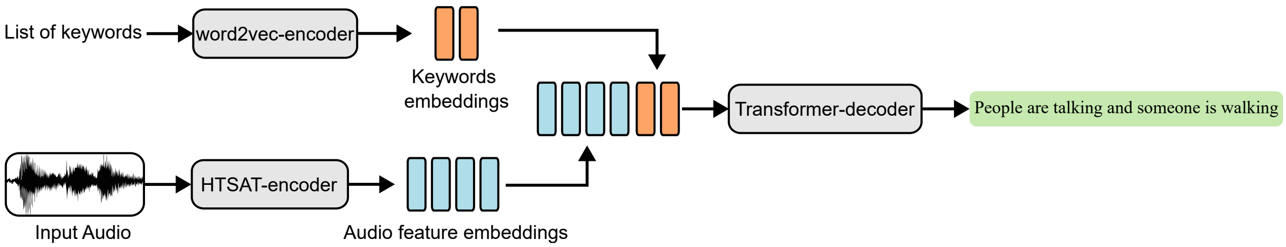


Figure 1: Guided captioning: keywords and audio are provided to the captioning system to produce a caption that is focused on the specific event given as keyword.

produce different captions depending on the provided guidance keyword, resulting in a significantly diverse set of captions compared to a system without keywords.

The paper is organized as follows: Section 2 introduces the concept of automatic audio captioning with keyword guidance, Section 3 presents the datasets and experimental setup; section 4 includes the discussion of the obtained results; section 5 presents conclusions and future work.

2. CAPTIONING WITH KEYWORDS GUIDANCE

The block diagram of the guided audio captioning system is presented in Fig 1. The system consists of two encoders, one for the keywords and another for the audio. The text encoder receives as input a list of keywords and will provide textual guidance to the model in the form of text embeddings; the audio encoder receives as input the raw audio signal to be transformed into audio feature embeddings. As a text encoder we trained a Word2Vec [16] model on the vocabulary of the dataset used in each experiment. To obtain the feature embeddings from the raw audio, we use the HTSAT transformer model [17] which is pre-trained on AudioSet.

The output of the text encoder, representing the keyword embeddings, is concatenated at the end of the audio embeddings obtained at the output of the audio encoder, forming the input for the transformer-decoder. The transformer-decoder has a standard architecture, as in [14], and is followed by a fully connected linear layer that outputs word probability. It has two hidden layers with a dimension of 768 and uses GELU activation functions in the feed-forward process between the hidden layers. The output of the transformer generates the captions based on the combined information from text and audio. The vocabulary used for training the model was collected from the reference captions for each dataset separately. KeyBERT [18] was used to extract N keywords for each clip, representing the words that best describe the reference captions.

The model was trained from scratch as opposed to using a pre-trained model, to accommodate for the concatenation of text embeddings and audio embeddings before decoding. For testing the model using textual guidance, we used two different setups: (1) using N keywords at once for guidance, same as in the training; and (2) using one keyword at a time. For the second setup, each clip is tested multiple times, each time with a different keyword. The keywords used in testing are obtained from the reference captions using the same procedure as for training, therefore they represent correct acoustic content for each clip.

3. EXPERIMENTAL SETUP

We use three datasets for our experiments, Clotho [3], collected based on Freesound [19] content, MACS [5], which contains audio clips of everyday acoustic environments, and AudioCaps [4], a subset of AudioSet [20].

Clotho contains 5929 recordings of 15 to 30 seconds long, each audio clip having five reference captions. We extract five keywords from the captions using KeyBERT ($N = 5$). The experiments are run on the development set of Clotho using the provided training/validation/test split. MACS contains 3930 recordings from TAU Urban Acoustic Scenes 2019 development dataset, from three acoustic scenes, each file being 10-seconds long. Captions and tags were collected at the same time for the data. A list of tags was provided to annotators to indicate what sounds they hear in the clip, after which they were asked to provide a one-sentence description. Here, we can use the tags provided by annotators as keywords (so N varies from 1 to 7 per clip). The experimental split is created based on the TAU Urban Acoustic Scenes Development set, with the included clips. AudioCaps contains 51308 clips, of which only 46721 are available now[1]. From these, 886 clips are used for testing. Because AudioCaps annotators had access to the AudioSet tags, we have tags available for the clips and can use them to guide the AAC system.

When using tags as keywords, it is important to note that: (1) the tags are not necessarily keywords that are extracted from the captions; (2) the tags can be single words (music) or compound terms (dog barking); (3) the number of tags per clip varies, so in this case N words provided as guiding text will be the number of tags for each clip. For Clotho we use $N = 5$ for all clips.

[1] Audio clips downloaded June 2024.

Training keywords	Guidance keywords	BLEU_1	BLEU_4	CIDEr	SPIDEr	% exact	% synonym	unique captions
None	None	56.24	15.19	39.35	26.21	-	-	737
kBERT 1	kBERT 1	58.99	17.14	47.02	30.22	47.08	11.77	774
kBERT 5	kBERT 5	66.13	20.65	63.64	**40.16**	44.30	11.40	944
kBERT 1	1 (all)*	57.73	16.38	41.31	27.14	40.52	10.37	2463
kBERT 5	1 (all)*	56.85	14.30	39.35	25.98	**72.94**	3.50	**3673**

* Five keywords extracted with kBERT for a clip are provided as guidance one at a time.

Table 1: Guided captioning results on CLOTHO dataset for different training and test setups: baseline (no keywords) and using 5 keywords extracted with keyBERT (kBERT). The main setup of the guided captioning system is highlighted with light gray.

The datasets differ on the number of unique captions (Clotho: 29614, MACS: 10594, AudioCaps: 47737), and the lexical diversity of the datasets also varies. The moving average type-to-token ratio (MATTR) [21] using a window of 500 tokens is 0.385 for Clotho, 0.302 for MACS and 0.415 for AudioCaps, indicating a richer vocabulary for the latter.

The main setup of the proposed system is to train it with the available N keywords per clip, and test it with one keyword as guidance. Each test audio clip is repeatedly tested with different keywords, and the produced captions are evaluated independently. This is marked in the tables in gray. As an ablation study, we compare the results with different setups. We first construct a baseline system as a plain AAC system using the same architecture but trained and tested without any keywords or guidance. We also train and test the system with only one keyword per clip, and train and test with N keywords at once. For MACS and AudioCaps, the ablation experiment also includes using for guidance all available tags per clip (variable N) in addition to the experiment with $N = 5$ keywords extracted with KeyBERT.

4. RESULTS AND DISCUSSION

The results of the system on Clotho are presented in Table 1. The performance of the baseline (None/None combination, on row 1), are aligned with the performance presented in the DCASE Challenge, placing the system around 6th place in the 2023 challenge. Training and testing with one keyword results in a significantly higher CIDEr and SPIDEr than of the baseline, which is further markedly improved when the system is trained and guided with 5 keywords at the same time. When the guidance goes through all keywords one at a time (lower half in Table 1), the system performs comparable with the baseline which does not use any guidance. However, in this experimental setup there are 5 times more test cases, because each clip is tested 5 times (once with each keyword). The advantage brought by using the most representative keyword per clip is lost when the averaging is done over all keywords, since there is more variety in the n-grams content of the predictions. Similarly, there is much less overlap in n-grams between captions containing one keyword compared to (potentially) five.

However, if we look at the generated captions, we observe that with different keywords the system produces a much higher number of unique sentences. To quantify the effect of the keywords guidance, we include to Table 1 the % of the times the generated caption contains the exact match of the guidance keyword or a synonym of it, respectively. When guided with 5 keywords, the % exact is calculated as the proportion of keywords present in the caption (so 1 of 5 counts as 20%). The keywords are most often present in the predicted caption exactly as such, rather than a synonym, due to the limited vocabulary of the system.

Results for AudioCaps and MACS are presented in Table 2. Ablations include the use of tags and KeyBERT produced keywords as guidance (none and five). When using tags, the number of keywords is equal to the number of tags available for each clip. While the numbers differ, the behavior is similar to what we observed on Clotho: guidance with five keywords at test time gives the best AAC metrics performance, while training with five and guiding with one keyword has similar AAC performance as the baseline (no guidance) but a much higher number of unique sentences. Particularly, for the case of AudioCaps, we achieve a SPIDEr score of 62.43% with 875 unique captions for 886 test audio files. Guiding the captioning process with a single keyword results in better scores when using tags rather than the KeyBERT keywords, but produces more repetitive captions, as shown by the smaller number of unique sentences. For MACS, the difference is not significant in CiDEr and SPIDEr score, likely due to the reduced lexical diversity and smaller vocabulary than the other datasets.

Table 3 provides a few examples of captions generated by the different setups for a clip in Clotho. It is evident that the use of keywords results in sentences containing the provided keywords. While the baseline produces a caption containing as much information as possible, the guided captions refer to different aspects of the environment through the keywords:

Dataset	Training keywords	Guidance keywords	BLEU$_1$	BLEU$_4$	CIDEr	SPIDEr	% exact	% synonym	unique captions
AudioCaps	None	None	69.92	27.74	72.50	45.39	-	-	608
	Tags	Tags	71.59	28.47	77.48	48.17	47.50	15.10	612
	kBERT	kBERT	86.82	33.17	102.4	**62.43**	75.28	10.34	**875**
	Tags	1 (all)*	70.80	26.90	69.04	43.65	46.82	13.26	1086
	kBERT	1 (all)*	51.08	14.50	36.66	23.62	96.27	0.11	1325
MACS	None	None	73.38	22.27	29.78	22.87	52.61	2.67	235
	Tags	Tags	75.61	24.83	32.43	24.44	53.75	2.77	173
	kBERT	kBERT	73.36	24.66	40.49	**28.77**	43.02	7.50	**523**
	Tags	1 (all)*	75.10	24.06	28.71	22.16	51.30	3.58	441
	kBERT	1 (all)*	69.51	20.32	29.86	22.41	48.60	4.80	1301

* All keywords for a clip are provided as guidance one at a time.

Table 2: Guided captioning results on AudioCaps and MACS datasets for different training and test setups: baseline (no keywords), using metadata labels (Tags) and using 5 keyBERT extracted labels as keywords (kBERT).

Keyword	Generated Caption
-	an announcement is made over a loudspeaker while people are talking in the background
crowded	people are talking in a **crowded** area and walking in a crowded area
restaurant	people are talking and moving in a **restaurant**
crowd	a **crowd** of people are talking in an enclosed space
busy	people are talking in a **busy** area with each other in the background
eating	a person is **eating** something and people are talking in the background

Table 3: Example captions generated for *je_PittsPhipps.wav* file in the CLOTHO dataset: baseline (no keywords) and guidance with 5 different keywords. Two of five reference captions for this clip contain the word “restaurant”.

scene (restaurant), attributes (busy, crowded), sound sources (crowd, eating). When evaluated with the captioning metrics, the caption produced by the baseline has the potential to being scored higher than the others due to containing more n-grams. On the other hand, there are specific terms, in this example “restaurant”, not picked up by the baseline. This is a good example of guidance, the focus on specific content instead of producing a generally good description. However, the guided captions quite often contain repeated keywords; the system likely requires a more careful optimization of the training process w.r.t. the length of the generated sentences. In this work, we kept the training process the same for all the scenarios, not optimizing them separately.

To verify the effect of random keywords on the guided captioning system, we feed as guidance five keywords that are not related to the content of the clip. The SPIDEr scores for the three datasets with this setup are 18.7 for Clotho, 10.7 for AudioSet and 20.5 for MACS, all smaller than the equivalents that are guided with correct keywords (the kBERT/kBERT line in the tables). Furthermore, if the system is not provided any keyword at test time, its SPIDEr scores are 18.8, 21.3 and 20.5, respectively, showing that the guidance does have a quantifiable effect on the system output.

5. CONCLUSIONS

This paper presented a guided captioning system to enhance the relevance of certain audio events in the generated caption. As a design choice, the system is not optimized for typical AAC metrics, and instead it focuses on user-provided keywords, which the AAC metrics fail to adequately evaluate. Because describing audio content is subjective to the annotator perception of the acoustic environment, there may be multiple correct ways to describe the same content; AAC metrics evaluate the largest overlap and penalize automatic captions with lesser content. Our focus on directing the system towards user-requested events intends to reduce this requirement. We demonstrated the system’s capability to produce a diverse set of descriptions aligned with the provided keyword. Future work will focus on better evaluating the captioning outputs based on the guidance keyword, since finding matching n-grams is not sufficient, nor necessary.

6. REFERENCES

[1] X. Mei, X. Liu, M. D. Plumbley, and W. Wang, “Automated audio captioning: an overview of recent progress and new challenges,” *EURASIP journal on audio, speech, and music processing*, vol. 2022, no. 1, p. 26, 2022.

[2] B. Winter, *Sensory Linguistics: Language, perception and metaphor*, ser. Converging Evidence in Language and Communication Research. John Benjamins, Apr. 2019.

[3] K. Drossos, S. Lipping, and T. Virtanen, “Clotho: an Audio Captioning Dataset,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2020, pp. 736–740.

[4] C. D. Kim, B. Kim, H. Lee, and G. Kim, “Audiocaps: Generating captions for audios in the wild,” in *Proc. of the 2019 Conf. of the North American Chapter of the Assoc. for Computational Linguistics: Human Language Technologies, Volume 1*, 2019, pp. 119–132.

[5] I. Martín-Morató and A. Mesaros, “Diversity and bias in audio captioning datasets,” in *Proceedings of the 6th Workshop on DCASE*, Nov. 2021, pp. 90–94.

[6] X. Mei, C. Meng, H. Liu, Q. Kong, T. Ko, C. Zhao, M. D. Plumbley, Y. Zou, and W. Wang, “WavCaps: A chatGPT-assisted weakly-labelled audio captioning dataset for audio-language multimodal research,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pp. 1–15, 2024.

[7] F. Gontier, R. Serizel, and C. Cerisara, “Automated audio captioning by fine-tuning BART with AudioSet tags,” in *Proceedings of the 6th Workshop on DCASE*, Barcelona, Spain, November 2021, pp. 170–174.

[8] Z. Zhou, Z. Zhang, X. Xu, Z. Xie, M. Wu, and K. Q. Zhu, “Can audio captions be evaluated with image caption metrics?” in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 981–985.

[9] F. Gontier, R. Serizel, and C. Cerisara, “Spice+: Evaluation of automatic audio captioning systems with pre-trained language models,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[10] I. Martín-Morató, M. Harju, and A. Mesaros, “A summarization approach to evaluating audio captioning,” in *Proceedings of the 7th Workshop on DCASE*, Nov. 2022, pp. 116–120.

[11] E. Labbé, T. Pellegrini, and J. Pinquier, “Is my automatic audio captioning system so bad? SPIDEr-max: A metric to consider several caption candidates,” in *Proceedings of the 7th Workshop on DCASE*, Nancy, France, November 2022.

[12] E. G. Ng, B. Pang, P. Sharma, and R. Soricut, “Understanding guided image captioning performance across domains,” in *Proceedings of the 25th Conference on Computational Natural Language Learning*, A. Bisazza and O. Abend, Eds. Association for Computational Linguistics, Nov. 2021, pp. 183–193.

[13] Y. Koizumi, R. Masumura, K. Nishida, M. Yasuda, and S. Saito, “A transformer-based audio captioning model with keyword estimation,” in *Proc. Interspeech 2020*, 10 2020, pp. 1977–1981.

[14] X. Mei, X. Liu, H. Liu, J. Sun, M. Plumbley, and W. Wang, “Automated audio captioning with keywords guidance,” DCASE2022 Challenge, Tech. Rep., May 2022.

[15] X. Xu, M. Wu, and K. Yu, “Diversity-controllable and accurate audio captioning based on neural condition,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 971–975.

[16] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” in *International Conference on Learning Representations*, 2013.

[17] K. Chen, X. Du, B. Zhu, Z. Ma, T. Berg-Kirkpatrick, and S. Dubnov, “HTS-AT: A hierarchical token-semantic audio transformer for sound classification and detection,” in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 646–650.

[18] M. Grootendorst, “KeyBERT: Minimal keyword extraction with BERT.” 2020. [Online]. Available: https://doi.org/10.5281/zenodo.4461265

[19] F. Font, G. Roma, and X. Serra, “Freesound technical demo,” in *ACM Int. Conf. on Multimedia (MM’13)*. Barcelona, Spain: ACM, Oct. 2013, pp. 411–412.

[20] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, “Audio set: An ontology and human-labeled dataset for audio events,” in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 776–780.

[21] M. Covington and J. McFall, “Cutting the gordian knot: The moving-average type-token ratio (MATTR),” *Journal of Quantitative Linguistics*, vol. 17, pp. 94–100, 05 2010.

AUDIO CAPTIONING IN FINNISH AND ENGLISH WITH TASK-DEPENDENT OUTPUT

Irene Martín-Morató, Manu Harju, Annamaria Mesaros

Signal Processing Research Centre, Tampere University, Finland
{irene.martinmorato, manu.harju, annamaria.mesaros}@tuni.fi

ABSTRACT

Describing audio content is a complex task for an annotator; the resulting caption depends on the annotator's language, culture and expertise. In addition, physiological factors like vision impairment may affect on how the sound is perceived and interpreted. In this work, we explore bilingual audio captioning in Finnish and English. In connection with this study, we release the SiVi-CAFE dataset, a small-size dataset of Sighted and Visually-impaired Captions for Audio in Finnish and English, with a collection of parallel annotations for the same clips. We analyze briefly the differences between captions produced by sighted and visually-impaired annotators, and train a system to produce captions in both languages that also mimics the style of different annotator groups. Obtaining a CIDEr score of 34.75% and 28.75% on the English and Finnish datasets, respectively. Furthermore, the system is able to perform a tagging task, obtaining F-score of 79.73%.

***Index Terms*—** audio captioning, visually-impaired users, captioning dataset, tagging, Finnish language

1. INTRODUCTION

Automated Audio Captioning (AAC) is a relatively recent researched topic [1], with potential applications that include accessibility aids [2] and content indexing for search engines [3]. While AAC systems have primarily focused on generating captions in English, there is a general growing demand for personalized content in other languages. Recent years have seen development of multilingual methods for image captioning [4], and also a few studies on multilingual AAC [5, 6]. The mentioned multilingual AAC works use translated captions, in this case between Chinese and English [5] and French, German and English [6].

Multilingual AAC can be obtained by generating captions directly in the target language, or generating captions in English and automatically translating them to the target language. However, while generating captions in English and then translating them to other languages can be faster and more straightforward, some nuances, idiomatic expressions, or cultural references may not translate accurately. Authors of [6] show that direct captioning in the target language may capture specific language nuances better. However, this requires language-specific training data, which is not easily available. Instead, there is the option of translating training data from English to the target language, though the disadvantages remain as pointed out above. Creating training data for AAC is a complex problem. Each annotator brings their unique style, influenced by factors such as age, culture, and language. In general, native speakers tend to use more precise and expansive language compared to non-native speakers [7]. One complicating factor is that humans are used to using language to describe visual rather than other sensory information; this is evident in the fact that languages often have a more extensive vocabulary for describing visual experiences compared to auditory ones [8], and this may affect the quality and diversity of captions, particularly when produced in a second language. Moreover, the annotation procedure affects the reference data: providing additional hints to annotators who can strongly bias their wording, as shown in [9]. Other factors can also influence the way we describe sounds. For example, individuals with visual impairments naturally pay more attention to auditory cues in their daily lives, as they need to rely on different sensory cues to understand their surroundings. Studies show that there are differences in the assessment of soundscape between visually-impaired people (ViP) and non-visually impaired ones, in terms of soundscape pleasantness or quietness [10]. As a special category of users with a heightened awareness of auditory cues, we would expect that visually-impaired annotators create richer audio captions than normal sighted individuals.

Considering the potential applications for captioning, and in particular accessibility, we expect that the need for more personalized output will become an important driving factor in development of captioning systems. To understand the possibility of creating a single universal captioning system that can produce outputs of different styles and in different languages for different categories of users, we adopt the approach proposed in [11] that used a *task embedding* for training an AAC system with different datasets and conditioned it to produce an output in the style of the dataset. In this work, we investigate a multitask training and conditioning across different languages and captioning styles, including ViP users.

The main contributions of this work are as follows: (1) a study of differences in captioning between visually-impaired and normal sighted users, in Finnish language, and a comparison from a linguistic point of view to parallel data in English; and (2) a multitask system trained with different languages and styles: Finnish, English, visually-impaired, biased, and non-biased captions.

There are a few unique elements in this study. Firstly, the use of an agglutinative language, in this case Finnish, as a typologically distant language from English, brings an element of novelty and difficulty to both the system vocabulary and its evaluation. Secondly, to the best of our knowledge, this is the first study using visually-impaired subjects in captioning as a category of annotators. The work aims to understand if such captions bring any advantage for training AAC, assuming they are more detailed. In conjunction with the study, we have published a multi-way annotated dataset that includes captions in English and Finnish: two sets of Finnish captions, ViP and sighted, and two sets of English captions, biased and non-biased in terms of vocabulary. Additionally, the dataset provides

This work was supported by Jane and Aatos Erkko Foundation under grant number 200061, "Guided captioning for complex acoustic environments". The authors wish to thank CSC-IT Centre of Science Ltd., Finland, for providing computational resources.

translations (automatically translated) between Finnish and English for the different captions sets.

The paper is organized as follows: in Section 2 we present shortly the data collection process and we analyze the differences between different types of annotations, focused on the use of language between Finnish and English and ViP and non-ViP. In Section 3 we introduce the multitask model training procedure, while in Section 4 we present the experimental results and discussion. Finally, Section 5 presents the conclusions and future work.

2. CAPTIONS WITH DIFFERENT ANNOTATOR PROFILE

The aim of the data collection process for this study was to obtain a variety of textual description for the same audio clips, in order to study how inter-cultural and linguistic differences between users produce different captions. In addition, we collected data from ViP users to study how visual impairment affects the descriptions. We started from the existing MACS dataset and proceeded with additional annotation tasks that have different annotator profile. The annotation task was similar for everyone, and followed the methodology presented in [9]. Audio clips are 10 seconds long, and the annotation was completed using a web-based interface that provided the clips one by one to be played back and annotated. The annotation process could be paused and continued later by logging in to the web platform. The complete collection of captions is published under the name SiVi-CAFE (Sighted and Visually-impaired CAptions in Finnish and English)[1].

2.1. Four-way data annotation

MACS dataset contains 10-second clips of audio from everyday environments (airport, public square and park) that were annotated by university students in a way that facilitated introducing bias in the captions. Namely, annotators were first given a tagging task, being asked to indicate what sounds from a given list of 10 classes they can hear [9]; after this, they were asked to describe the clip in one sentence. The sentences were found to contain the exact wording of the tags for 41.78% of the sentences [9]. In the SiVi-CAFE collection, this set is referred to as *English-bias*.

The same setup was repeated with another pool of students, this time without the tagging task. In contrast with the observations on biasing, the captions produced in this setup have a larger vocabulary and longer average caption length. In the SiVi-CAFE collection, this set is referred to as *English-nobias*.

Finnish language data collection has focused on obtaining captions from visually-impaired users. The annotation was performed using a company that employs visually-impaired workers for various tasks. We recruited 25 persons, native Finnish speakers, through Aarnikukko Oy[2] and provided them with an accessible web-based tool for the annotation process. Of the 25 annotators with visual disability, 14 reported themselves as blind, 9 as partially sighted, and two did not answer. In addition, 11 participants announced to have some environmental perception via vision, including 3 of the blind individuals. Each worker annotated 180 clips, resulting in 900 clips each having 5 captions. We refer to this set as *Finnish-ViP*.

The same 900 clips were used to collect parallel data in Finnish from normal-sighted people using volunteers who were native speakers; this set of captions is referred to as *Finnish*. This subset is also incomplete, i.e. not all 900 clips have 5 captions.

Dataset	Audio clips	Unique sentences	Sentence length (std)	Vocab. size	MATTR (std)
English-bias	3930	16262	9.5 (3.89)	2717	0.26 (0.02)
English-nobias	2050	9679	10.2 (3.78)	2685	0.27 (0.02)
Finnish-ViP	900	4458	8.3 (3.25)	4518	0.39 (0.03)
Finnish	900	3592	7.5 (2.79)	3540	0.37 (0.03)

Table 1: Statistics of the collected datasets.

2.2. Analysis of the annotations

The main difficulty in data collection was recruiting sufficiently many annotators. Some tasks were implemented with student volunteers that received various rewards for their time (e.g. movie tickets). There was added difficulty in recruiting digitally fluent visually-impaired workers; for this reason the *Finnish-ViP* data is relatively small. Moreover, while the Finnish annotators are native speakers, the ones providing English annotations are international students using English in their studies, hence very likely not English native. As discussed earlier, this probably affects their use of language for describing the sounds. The *English-bias* data was produced by 133 annotators, *English-nobias* by 89, *Finnish-ViP* by 25, and *Finnish* by 42. The sets are each somewhat incomplete, but there are 3612 captions provided for 900 clips which were annotated by all categories of users, and can be considered as parallel data. For completeness, all original data was translated into the other language using the DeepL translation API[3], following [6].

The statistics of the different caption sets are provided in Table 1. To characterize the lexical diversity, we use the type-token ratio (TTR) the ratio between the unique words (types) and total words (tokens) in each set. To account for the difference in size, we calculate the moving average TTR (MATTR) [12] which calculates TTR every 500 words, hence MATTR allows comparing texts of different lengths. While the two languages are not comparable, the difference between *English-bias* and *English-nobias* shows a difference in lexical diversity, as does the *Finnish-ViP* compared to *Finnish*.

The 3612 captions that form a parallel corpus results in a vocabulary of 1132 and 1328 for the *English-bias* and *English-nobias* sets, respectively, while for *Finnish-ViP* and *Finnish* the vocabulary size is 2142 and 2498, respectively. The *Finnish-ViP* set has the richest vocabulary; this is also reflected in the high MATTR.

The most interesting detail is the way annotators describe the location of the sounds in the audio clips. While all groups indicated sounds as appearing in the background (Fi: taustalla), in the distance (etäällä), far away (kaukainen), or less often nearby (Fi: lähempänä, comp.), the visually impaired Finnish speakers described egocentric directions by indicating sounds being ‘on the right’ (oikealla) or ‘on the left’ (vasemmalla). In the *Finnish* set, ‘on the left’ appears 10 times and ‘from the left’ once, while in the *Finnish-ViP* set there are 443 variants for ’left’ (including ‘to the left’, ‘on the left’, ‘from the left’, ’front left’, ’back left’). Similarly, variations of ’right’ appear 397 times in the *Finnish-ViP* set, and only 8 times in the *Finnish* set. In the English data “on the left” appears 19 times and “on the right” only 14 times.

3. A UNIVERSAL CAPTIONING SYSTEM

A single model is trained using all the different annotation types, in order to create a universal captioning system. We employ a task embedding token as proposed in [11]; each different annotation type is seen as a task that is assigned a specific token. Figure 1 illustrates

[1] https://doi.org/10.5281/zenodo.11505823

[2] https://www.aarnikukko.fi/

[3] https://www.deepl.com/pro-api

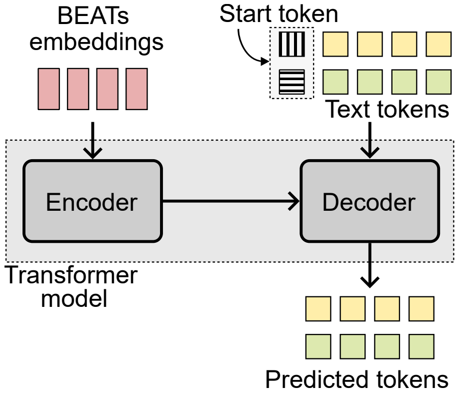


Figure 1: Block diagram of the AAC system with task tokens.

how a *start token* is concatenated at the beginning of the sentence for each of the different annotation types. The translated datasets are also considered as separate tasks, to provide the system with the ability of producing as large variety of styles as possible.

The model follows a standard transformer architecture and a pre-trained tokenizer. The tokenizer is based on Byte-Pair Encoding (BPE), and creates a list of unique words with their frequency; a vocabulary size parameter has to be selected beforehand. Before training the transformer, the first step is to fit the tokenizer. The role of the tokenizer is to split the sentences into words and then into subwords. Finally, those subwords are converted to ids using a look-up table; this will facilitate generation of words that have not been seen in the training vocabulary, achievable by breaking unknown words into smaller units that the tokenizer can recognize. The tokenizer is trained with the vocabulary of all the datasets, English and Finnish originals and the translated versions. The maximum vocabulary size is set to 5000, which was experimentally found to be sufficient to wrap English and Finnish language. For each dataset we use a *start token* as done in [11], indicating to which dataset the caption belongs to. The audio is fed to the model after a pre-processing step where a feature extractor is used. We use the pre-trained encoder BEATs [13] as feature extractor; the resulting embeddings are used as inputs to the transformer encoder. We chose BEATS as audio representation based on the system that achieved the best performance in the DCASE 2023 Challenge Audio Captioning task. However, to reduce the number of input tokens, we average pool the BEATs embeddings over the time dimension with a factor of 32.

3.1. Experimental setup and evaluation

The system is evaluated in a 10-fold manner, because the distribution of the data is unbalanced among datasets; the smaller dataset (*Finnish*) is used as norm for splitting the data into folds based on the 10 cities where the data has been recorded. We report results using BLEU [14] (a measure of n-gram overlap between generated and reference captions) and CIDEr-D [15] (consensus-based measure from image captioning), as language-agnostic measures. We also use sentence-BERT cosine similarity (sBERT$_{sim}$) [16] as a more meaning-oriented metric that compares the captions at sentence level. For the Finnish captions we calculate this metric using TurkuNLP_sbert [17], shown to perform better on the Finnish language tasks than a multilingual version; for English we use paraphrase-multilingual-mpnet-base-v2 as used in [6].

Dataset	BLEU$_1$	CIDEr	sBERT$_{sim}$
English-bias	45.65	20.95	60.15
English-nobias	48.40	21.61	60.13
Finnish-ViP	29.98	9.97	72.88
Finnish	26.80	12.38	74.64

Table 2: Human-to-human evaluation of captions. One caption is randomly selected as predicted and compared with the other captions available for the same clip.

3.2. Human-to-human evaluation

To analyze the connection between system predictions and human-produced annotations, we calculate the human-to-human comparison for the datasets using the same metrics. Their values for the original (annotated) data are presented in Table 2. Unigram overlaps, shown by BLEU$_1$, are strong for the English datasets and less for Finnish; based on CIDEr, *Finnish-ViP* has the least consensus in descriptions between annotators. $sBERT_{sim}$ is very similar for the English sets, while *Finnish-ViP* has a somewhat higher $sBERT_{sim}$ than Finnish, indicating that ViP annotations are more similar in meaning, even though their wording differs.

4. EXPERIMENTAL RESULTS

4.1. Captioning results

Table 3 shows the AAC metrics for the multitask model, including cross-testing in which we generate the caption with a specific task token, and evaluate against a different reference set of the same language. For the *English-bias* data, the model achieves a CIDEr score of 34.72%, which is, surprisingly, almost 14 points higher than the human performance. This can be attributed to the fact that the models typically generate rather repetitive captions, more so than the human annotators. We verify this by inspecting the top 3-grams: “in the background” appears 607 and 887 times in *English-bias* and *English-nobias*, respectively, while in the predicted outputs they appear 167 times and 372 times for the *English-bias* and *English-nobias* captioning style, respectively. The next most common 3-grams in the predicted captions are “talking and walking”, appearing 210 and 154 times, respectively; and “people are talking”, 71 and 235 times. For the *Finnish* dataset we achieve a CIDEr of 17.31%, while for *Finnish-ViP* we achieve a CIDEr of 13.88%, both higher than the human-to-human evaluation.

For comparison, we trained monolingual models as multitask models but using only the data from a single language, including the automatically translated captions from the other language. In general the monolingual models had a slightly worse performance, being trained with less data. For *English-bias* we achieve BLEU$_1$ 63.80%; CIDEr 33.95% and sBERT$_{sim}$ 60.36%, while for *Finnish*, we achieve BLEU$_1$ 42.97%; CIDEr 15.99% and sBERT$_{sim}$ 74.18%.

Comparing the predicted captions against reference captions with a different style produces lower scores, with a few exceptions: *Finnish* vs *Finnish-ViP* has a good BLEU$_1$, showing a high overlap in unigrams; all cross-evaluations for Finnish language have a very similar sBERT$_{sim}$, showing that the descriptions are similar in meaning, although not in the exact wording. English sets always score much lower when evaluated against another style.

Prediction	Reference	BLEU$_1$	BLEU$_4$	METEOR	CIDEr	sBERT$_{sim}$
English-bias	English-bias	67.07	18.32	20.60	34.72	61.64
English-bias	English-nobias	62.03	14.58	17.86	25.01	59.46
English-nobias	English-nobias	69.16	21.40	21.69	33.66	59.28
English-nobias	English-bias	58.64	13.80	18.66	26.04	57.63
Finnish-ViP	Finnish-ViP	51.30	4.85	14.63	13.88	73.73
Finnish-ViP	Finnish	43.64	4.54	13.50	15.17	74.01
Finnish	Finnish	46.29	5.28	14.02	17.31	75.04
Finnish	Finnish-ViP	50.30	5.55	13.90	13.63	74.40

Table 3: Results on all the datasets using the multitask model, with evaluation across same language reference sets.

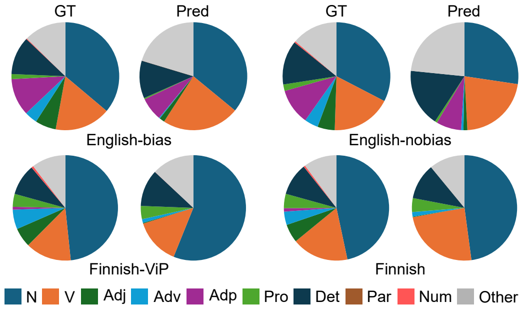


Figure 2: Pie charts showing the POS analysis of the English and Finnish datasets. N (Noun), V (Verb), Adj (Adjective), Adv (Adverb), Adp (Adposition), Pro (Pronoun), Par (Particle), Num (Numeral). GT stands for Ground truth; Pred stands for predicted captions from the multitask model.

4.2. Generated language analysis

We investigate the sentence structure in the reference and predicted captions by performing a Part-Of-Speech (POS) analysis and visualize the proportions of POS as pie charts in Fig. 2; for English we use the spaCy[4] toolbox, for Finnish the Finnish-tagtools software[5]. We easily notice that captions are mainly formed using nouns and verbs, with nouns dominating the sentences; the reference annotations also contain a non-negligible percentage of adverbs and adjectives. The charts show a clear difference between the languages: English datasets make more use of verbs, prepositions and determinants, while the Finnish datasets use more nouns, adverbs and pronouns. The predicted captions on the other hand contain almost no adverbs or adjectives, which is an interesting observation that holds for both languages. The system produces a good proportion of adpositions for English and pronouns for Finnish, but overall the model is mostly generating nouns and verbs. The difference between *English-bias* and *English-nobias* is reflected in the predicted captions: “adults talking”, “children voices” and “birds singing” are mentioned 154, 58 and 183 respectively for *English-bias* style, while they do not appear in this exact form at all in *English-nobias*. This comes from the training data, where they appear 2179, 443 and 1241 times, and only 20, 1 and 113 times in *English-bias* and *English-nobias*, respectively.

4.3. Tagging system

As a different annotation type indicated by the *start token*, it is also possible to use the multitask model as a tagging system. In this case, instead of the caption, the system receives in training the concatenated tags, seen as a sentence, though it is not a grammatically correct one. The *English-bias* dataset has tags available that were collected during the same annotation process as the caption, as explained in [9]. With the task token we indicate that we require similar “sentences”. The model achieves an overall micro-F1 score of of 79.73% (Precision 77.01% and Recall 82.64%) for tagging.

GT tags	Predicted caption
adults talking, traffic noise, music	“**music** is playing and people are talking”
children voices, footsteps	“birds singing and **children voices**”
birds singing, traffic noise	“**traffic noise** and **birds singing**”
adults talking, footsteps	“people are talking and walking”

Table 4: Examples of *English-bias* predicted captions and the reference tags for the respective clips; tags exact matches are in bold.

Tags also allow evaluating if the predicted captions match the sound events tagged in the reference for each clip. If we evaluate the predicted *English-bias*-style captions against the reference tags as captions, we obtain BLEU$_1$ 27.66%, CIDEr 17.07% and sBERT$_{sim}$ 67.14%; for *English-nobias*-style captions BLEU$_1$ is 10.40%, CIDEr is 4.16% and sBERT$_{sim}$ is 57.40%. This evaluation setup illustrates well the induced bias, i.e. the annotators being hinted the tags while listening the clip for recognizing the sounds.

Finally, we calculate to what extent the reference tags are present in the predicted captions, obtaining that 51.3% of the predicted captions with the *English-bias* task token have at least one correct n-gram. A few examples are shown in Table 4. Only exact matches can be easily identified; however, we can observe that captions may contain very similar words to the tags, e.g.“people are talking” matching in meaning “adults talking” in the provided examples.

5. CONCLUSIONS

This paper presented a more linguistically-oriented study to AAC, focusing on a parallel corpus of linguistically-different references. The work introduced a dataset comprised of captions in English and Finnish, including annotations provided by visually-impaired users. We designed a multitask system that can produce captions in all required styles, including tags. The dataset analysis shows differences between languages and user types, which were well modeled by the proposed method. Most importantly, the proposed captioning system was capable to learn from a collection of tasks that share some information, i.e. the audio content, but are at the same time very different, i.e. the language or style. We have also successfully shown that the system can be combined with more simplified tasks, in this case audio tagging, paving the way for developing linguistically-mixed systems that can handle multiple languages and multiple sentence styles.

[4] https://spacy.io/

[5] http://urn.fi/urn:nbn:fi:lb-2021101101

6. REFERENCES

[1] X. Mei, X. Liu, M. D. Plumbley, and W. Wang, “Automated audio captioning: an overview of recent progress and new challenges,” *EURASIP journal on audio, speech, and music processing*, vol. 2022, no. 1, p. 26, 2022.

[2] O. Alonzo, H. V. Shin, and D. Li, “Beyond subtitles: Captioning and visualizing non-speech sounds to improve accessibility of user-generated videos,” in *Proceedings of the 24th International ACM SIGACCESS Conference on Computers and Accessibility*, ser. ASSETS ’22. New York, NY, USA: Association for Computing Machinery, 2022.

[3] A. S. Koepke, A.-M. Oncescu, J. F. Henriques, Z. Akata, and S. Albanie, “Audio retrieval with natural language queries: A benchmark study,” *IEEE Transactions on Multimedia*, vol. 25, pp. 2675–2685, 2022.

[4] R. Ramos, B. Martins, and D. Elliott, “LMCap: Few-shot multilingual image captioning by retrieval augmented language model prompting,” in *Findings of the Association for Computational Linguistics: ACL 2023*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 1635–1651.

[5] M. Wu, H. Dinkel, and K. Yu, “Audio caption: Listen and tell,” in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 830–834.

[6] M. Cousin, E. Labbé, and T. Pellegrini, “Multilingual Audio Captioning using machine translated data,” Sept. 2023, working paper or preprint. [Online]. Available: https://hal.science/hal-04220315

[7] C. Bentz, A. Verkerk, D. Kiela, F. Hill, and P. Buttery, “Adaptive communication: Languages with more non-native speakers tend to have fewer word forms,” *PloS one*, vol. 10, no. 6, p. e0128254, 2015.

[8] B. Winter, “Sensory linguistics,” *Converging Evidence in Language and Communication Research*, 2019.

[9] I. Martn-Morató and A. Mesaros, “Diversity and bias in audio captioning datasets,” in *Proceedings of the 6th Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCASE2021)*, Barcelona, Spain, November 2021, pp. 90–94.

[10] J. Vida, J. A. Almagro, R. García-Quesada, F. Aletta, T. Oberman, A. Mitchell, and J. Kang, “Urban soundscape assessment by visually impaired people: First methodological approach in granada (spain),” *Sustainability*, vol. 13, no. 24, 2021.

[11] E. Labbé, T. Pellegrini, and J. Pinquier, “CoNeTTE: An efficient Audio Captioning system leveraging multiple datasets with Task Embedding,” Sept. 2023, working paper or preprint. [Online]. Available: https://ut3-toulouseinp.hal.science/hal-04193791

[12] M. Covington and J. McFall, “Cutting the gordian knot: The moving-average type-token ratio (MATTR),” *Journal of Quantitative Linguistics*, vol. 17, pp. 94–100, 05 2010.

[13] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, Z. Chen, W. Che, X. Yu, and F. Wei, “BEATs: Audio pre-training with acoustic tokenizers,” in *International Conference on Machine Learning*. PMLR, 2023, pp. 5178–5193.

[14] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “BLEU: A method for automatic evaluation of machine translation,” in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ser. ACL ’02. USA: Association for Computational Linguistics, 2002, p. 311–318.

[15] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, “CIDEr: Consensus-based image description evaluation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4566–4575.

[16] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence embeddings using Siamese BERT-networks,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 3982–3992.

[17] J. Kanerva, F. Ginter, L.-H. Chang, I. Rastas, V. Skantsi, J. Kilpeläinen, H.-M. Kupari, J. Saarni, M. Sevón, and O. Tarkka, “Finnish paraphrase corpus,” in *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa’21)*. Linköping University Electronic Press, Sweden, 2021, pp. 288–298.

ACOUSTIC SCENE CLASSIFICATION ACROSS MULTIPLE DEVICES THROUGH INCREMENTAL LEARNING OF DEVICE-SPECIFIC DOMAINS

Manjunath Mulimani, Annamaria Mesaros

Signal Processing Research Center, Tampere University, Tampere, Finland
{manjunath.mulimani, annamaria.mesaros}@tuni.fi

ABSTRACT

In this paper, we propose using a domain-incremental learning approach for coping with different devices in acoustic scene classification. While the typical way to handle mismatched training data is through domain adaptation or specific regularization techniques, incremental learning offers a different approach. With this technique, it is possible to learn the characteristics of new devices on-the-go, adding to a previously trained model. This also means that new device data can be introduced at any time, without a need to retrain the original model. In terms of incremental learning, we propose a combination of domain-specific Low-Rank Adaptation (LoRA) parameters and running statistics of Batch Normalization (BN) layers. LoRA adds low-rank decomposition matrices to a convolutional layer with a few trainable parameters for each new device, while domain-specific BN is used to boost performance. Experiments are conducted on the TAU Urban Acoustic Scenes 2020 Mobile development dataset, containing 9 different devices; we train the system using the 40h of data available for the main device, and incrementally learn the domains of the other 8 devices based on 3h of data available for each. We show that the proposed approach outperforms other fine-tuning-based methods, and is outperformed only by joint learning with all data from all devices.

Index Terms— Domain-incremental learning, Low-Rank Adaptation, Batch Normalization, acoustic scene classification, mismatched devices

1. INTRODUCTION

Deep learning models have recently shown impressive results for acoustic scene classification (ASC) tasks from in-domain static data. However, in realistic scenarios, new data comes in sequentially. This new data may be from a different domain than the data used to optimize the model. Incremental or continuous learning of such a sequence of mismatched domains (i.e., locations, devices, or other acoustic conditions) deteriorates the model performance on previously learned domains when learning a new one, which means catastrophic forgetting [1] occurs in the absence of the previous domain's data. Mismatched conditions in continuously evolving domains introduce domain shift or bias in the feature distribution, which is the main reason for performance degradation.

In this work, we propose to use the domain-incremental learning (DIL) [2] approach for learning ASC tasks from different domains (devices) without forgetting the acoustic scenes from previously seen domains.

DIL was successfully applied to detect objects from road scenes in different locations [3] and in different weather conditions [2] for images, and acoustic scenes from different locations [4]. We aim to develop a practical DIL model to effectively classify acoustic scenes from all recording devices seen so far by going through the stream of data only once, in online learning mode.

DIL is different than existing domain adaptation (DA) methods for ASC from different devices [5–7]. DA setup typically includes two domains: source and target. It transfers the knowledge from the source to the target domain and only focuses on the accuracy of the target domain. DA requires access to the data of the source domain to match the distribution with the target domain. In comparison to DA, the DIL setup includes multiple domains over time that the system needs to adapt to; it focuses on the overall accuracy of all the domains seen so far; takes additional measures to alleviate the forgetting; and typically does not have access to the previous domain's data.

Our previous work adapts the model for the new locations sequentially by updating only the running statistics i.e., running mean and variance of BN layers in an online domain incremental learning (ODIL) setup [4]. In this work, we propose to add Low-Rank Adaptation (LoRA) parameters to the convolutional layers of the model, and update only these LoRA parameters and running statistics of the BN layers to adapt to the incrementally occurring new devices for effective ASC. LoRA is a parameter-efficient fine-tuning (PEFT) method widely used as a fine-tuning strategy for transformer-based Large Language Models (LLMs) [8]. LoRA is also used with vision transformers for continual learning of images [9] and also applied to convolutional layers for DA [10] and segmentation [11] of images.

The use of LoRA with CNN-based models for ODIL in the context of audio devices is yet to be explored. Unlike conventional fine-tuning, in which all the parameters of the model are updated to adapt to a new domain, LoRA fixes the other parameters of the current model and only updates the trainable low-rank matrices on the new domain, sequentially. LoRA parameters are significantly less than the total parameters of the original model.

The main contributions of this work are as follows,

- We propose using LoRA parameters for ODIL to learn acoustic scenes incrementally from mismatched devices.
- We investigate the combination of LoRA and BN statistics in classifying acoustic scenes in both online and offline settings.
- We also investigate the ability of the proposed approach trained on a device with enough data to adapt to incoming mismatched devices with limited data. It verifies the suitability of LoRA in low-data scenarios.

The rest of the paper is organized as follows: Section 2 presents the notations, baselines, and the proposed LoRA and BN combination

This work was supported by Jane and Aatos Erkko Foundation grant 230048 "Continual learning of sounds with deep neural networks". The authors wish to thank CSC-IT Centre of Science Ltd., Finland, for providing computational resources.

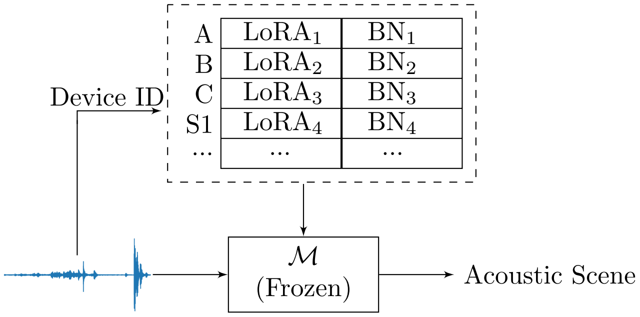


Figure 1: Overview of the proposed approach for incremental learning of acoustic scenes from different devices in sequence. Inputs to the model are the test sample and the device ID. The frozen model $\mathcal{M}$ uses domain-specific LoRA parameters and BN statistics to classify the acoustic scenes from a particular device such as A, B, C, S1, and so on.

for ODIL of acoustic scenes. Section 3 introduces the datasets, implementation details, and results. Finally, conclusions are given in Section 4.

2. INCREMENTAL LEARNING OF DEVICE DOMAINS

2.1. Incremental learning setup and notations

In our incremental learning setup, a sequence of ASC tasks is presented to the model; these tasks represent the datasets from different domains: $\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_t$. The model learns each task, i.e., $\mathcal{D}_t$ in our case, at incremental time step t. A domain $\mathcal{D}_t$ is an acoustic scene dataset recorded with a particular device, composed of audio clips and corresponding class labels. All domains share the same classes. We aim to train a single-model $\mathcal{M}$ that learns to classify the same acoustic scenes when domain or data distribution changes. Initially, we train the $\mathcal{M}$ on a relatively larger dataset $\mathcal{D}_1$ offline and this model is a base model for incremental tasks. During the training of incremental tasks, $\mathcal{M}$ follows a realistic setting where it sees a stream of samples only once, online, and quickly adapts to the new domain on the fly, i.e., ODIL. More importantly, the performance of the $\mathcal{M}$ does not degrade on previous domains when it learns a new domain, unlike the domain adaptation case, in which the performance on the previous domain does not matter. Note that in this work we refer to $\mathcal{D}_t$ as task, domain, device, and dataset interchangeably.

2.2. Baselines

We construct a few standard baselines to compare with the proposed approach: (1) *Feature extraction (FE)*: the feature extractor component of the base model is frozen after learning $\mathcal{D}_1$. The classifier is updated in each incremental domain; (2) *Conventional Finetuning (FT)*: a model trained on the previous domain is fine-tuned on the new domain at each incremental time step with all its parameters. The model is being trained incrementally; (3) *Disjoint*: a base model is trained separately on each domain. (4) *Joint*: a base model is retrained from all the data of the domains seen so far in each incremental time step, breaking one of the constraints of the DIL. For a fair comparison, the base model on $\mathcal{D}_1$ is trained offline and on other domains trained online in incremental steps for all methods.

2.3. Online domain-incremental learning of devices using LoRA-BN combination

We propose to compute domain-specific LoRA parameters and BN statistics for ODIL. At the initial time step $t = 1$, the base model $\mathcal{M}$ is trained on dataset $\mathcal{D}_1$. At each incremental time step i, $\mathcal{M}$ is frozen and we only update its LoRA parameters and BN statistics using new dataset $\mathcal{D}_i$ as explained below.

Low-Rank Adaptation parameters

For a weight matrix $\boldsymbol{W}_{base} \in \mathbb{R}^{m \times n}$ of a convolutional layer of the base model $\mathcal{M}$, LoRA adds trainable rank decomposition matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ as:

$$W_{base} + \Delta W = W_{base} + AB, \tag{1}$$

where $A \in \mathbb{R}^{m \times r}$ is a down-projection matrix, $B \in \mathbb{R}^{n \times r}$ is a up-projection matrix and rank r is much smaller than the size of the inputs m and outputs n, i.e., $r \ll min(m, n)$. The forward pass of the network with LoRA changes from $\boldsymbol{W}_{base}\boldsymbol{x}$ to:

$$\boldsymbol{h} = \boldsymbol{W}_{base}\boldsymbol{x} + \boldsymbol{AB}\boldsymbol{x}, \tag{2}$$

where $\boldsymbol{x}$ is the input and $\boldsymbol{h}$ is the hidden output. During incremental learning of a new domain, $\boldsymbol{W}_{base}$ is frozen and only the domain-specific weights of $\boldsymbol{A}$ and $\boldsymbol{B}$ are updated and stored in the model.

Statistics of Batch Normalization layer

BN normalizes the input activations of each layer using mini-batch statistics, i.e., running mean and variance. The behavior of the BN layer is different in the training and inference phases. During training, statistics of the BN layer are updated using training data forwarded through the network. During inference, statistics obtained from the training phase are fixed and used to standardize each layer of the network. BN performs well only when training and testing data come from the same domain. Therefore, we compute statistics for each domain separately and store into the model during training.

During inference at each incremental time step, domain-specific LoRA parameters and BN statistics are applied to base model $\mathcal{M}$ to classify acoustic scenes from the current domain, as shown in Fig. 1. Input to the model is a combination of the device ID and test sample, similar to task-incremental learning [12]. Device ID locates the LoRA parameters and BN statistics of the corresponding device before classifying the test sample. We only update additional LoRA parameters and statistics of the BN layers; all other parameters of the $\mathcal{M}$ are fixed. This allows us to recover the original performance of $\mathcal{M}$ for each device by replacing the corresponding LoRA parameters and BN statistics. Therefore, $\mathcal{M}$ does not suffer from forgetting previous devices when it learns a new device. Hereafter, we refer to our proposed approach as LoRA-BN.

3. EVALUATION AND RESULTS

3.1. Dataset and training setup

Experiments are conducted on the TAU Urban Acoustic Scenes 2020 Mobile development dataset [13], containing audio recordings from 3 real devices: denoted as A, B, and C, and additional S1-S6 devices simulated from device A. The domain $\mathcal{D}_1$ is composed of 40 hours of audio data from device A; the other 8 domains $\mathcal{D}_2$ to $\mathcal{D}_9$ include 3 hours of data each from devices B, C and S1-S6. We

Table 1: Device-specific accuracy of the different methods on each current domain.

Method	$\mathcal{D}_1$ A	$\mathcal{D}_2$ B	$\mathcal{D}_3$ C	$\mathcal{D}_4$ S1	$\mathcal{D}_5$ S2	$\mathcal{D}_6$ S3	$\mathcal{D}_7$ S4	$\mathcal{D}_8$ S5	$\mathcal{D}_9$ S6
Base	67.2	37.2	36.1	19.1	18.9	21.7	26.6	23.5	22.4
ODIL-BN [4]	67.2	40.8	44.7	23.1	22.5	26.0	30.5	31.2	25.4
LoRA-BN	67.2	**47.0**	**52.3**	**37.0**	**37.4**	**39.7**	**42.4**	**43.3**	**34.6**

Table 2: Average accuracy of the different methods over current and all previously seen domains.

Method	$\mathcal{D}_1$ A	$\mathcal{D}_2$ B	$\mathcal{D}_3$ C	$\mathcal{D}_4$ S1	$\mathcal{D}_5$ S2	$\mathcal{D}_6$ S3	$\mathcal{D}_7$ S4	$\mathcal{D}_8$ S5	$\mathcal{D}_9$ S6
FE	67.2	46.5	43.6	33.3	24.7	30.3	33.6	33.6	34.0
FT	67.2	48.0	48.4	37.7	33.1	39.0	43.9	43.4	44.0
Disjoint	67.2	48.0	46.6	35.3	29.1	35.9	36.2	34.9	36.9
LoRA-BN	67.2	**57.1**	**55.5**	**50.9**	**48.2**	**46.8**	**46.2**	**45.8**	**44.7**
Joint	67.2	60.3	59.7	56.3	56.1	55.0	56.4	56.7	54.7

follow the official training and testing split provided in the dataset to generate the data for each domain/device[1].

Initially, the model is trained on the domain $\mathcal{D}_1$ and it adapts to the remaining domains in incremental time steps. We follow the standard procedure in incremental learning, where the model is only trained on the current domain, without any data from previous domains, and evaluated on all previously seen domains.

3.2. Implementation details and evaluation metrics

We use the 6 convolutional blocks as a feature extractor and the layers specifications of each block are the same as PANNs CNN14 [14]. The global pooling is applied to the last convolutional layer to get a fixed-length input feature vector to the classifier. The entire network is trained from scratch on the first domain $\mathcal{D}_1$ as the base model. This base model is adapted to the other domains in incremental time steps. Input audio recordings are resampled to 32 kHz and log mel spectrograms are computed using default settings provided in [14].

The model is trained using the Adam optimizer [15] with a learning rate of 0.0001 and a mini-batch size of 32. The number of epochs to train the model on $\mathcal{D}_1$ is set to 120. The LoRA-BN and baselines are trained at incremental time steps for one epoch only. CosineAnnealingLR [15] scheduler updates the optimizer in every epoch. The rank r is set to 2 for minimal trainable parameters and the original kernel weight is 3.

We evaluate the performance of the model on the current domain and all previously seen domains at each incremental step using average accuracy and forgetting (Fr) as defined in [4]. Average accuracy is the average of accuracies of the method over the current and all previously seen domains. Average forgetting (Fr) is the average difference between the accuracy of the model for each domain at its learning iteration (the first time the model learns this domain) and the accuracy of the model for the same domain at the current iteration (after learning the current domain). A higher average accuracy and lower Fr are better.

3.3. Results

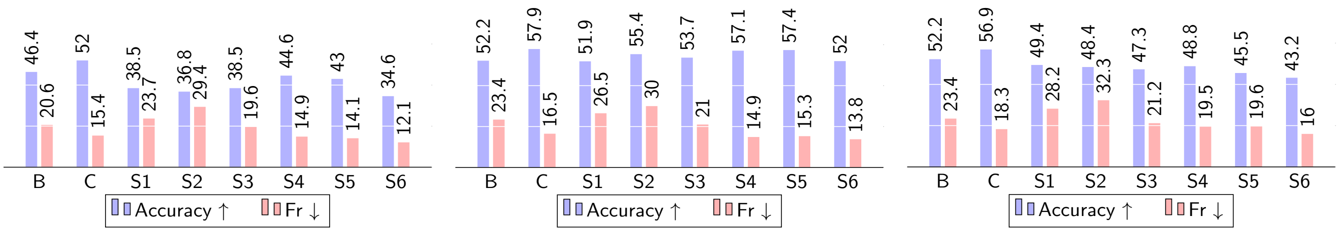
The base model trained on data from real device A achieved an accuracy of 67.2% for domain $\mathcal{D}_1$. In Table 1, we compare the accuracy of proposed LoRA-BN on the current domain with other methods, in which all the parameters are frozen or only a few device-specific parameters are updated in incremental steps and therefore not suffer from forgetting.

To check the severity of the mismatch between domain $\mathcal{D}_1$ and other incremental domains $\mathcal{D}_2$ to $\mathcal{D}_9$, we use the base model to classify the acoustic scenes of other domains without updating its parameters (no training). The base model does not adapt to the incremental domains, resulting in a drastic performance drop, especially from simulated domains, $\mathcal{D}_4$ to $\mathcal{D}_9$, as seen in Table 1.

ODIL-BN computes the domain-specific running statistics of the BN layers to classify acoustic scenes from each domain [4]. ODIL-BN does not change any other parameters of the base model and does not forget previous domains. However, this alone improves the performance of the base model only slightly in most of the incremental domains. The proposed LoRA-BN computes the domain-specific LoRA parameters for each convolutional layer and domain-specific running statics for each BN layer. The additional combined LoRA parameters and running statistics help the base model to effectively adapt to the incremental domains. It can be seen that LoRA-BN improves the performance for $\mathcal{D}_2$ by 9.8%p (percentage point), $\mathcal{D}_3$ by 16.2%p, $\mathcal{D}_4$ by 17.9%p, $\mathcal{D}_5$ by 18.5%p, $\mathcal{D}_6$ by 18.0%p, $\mathcal{D}_7$ by 15.8%p, $\mathcal{D}_8$ by 19.8%p, $\mathcal{D}_9$ by 12.2%p, compared to the base model.

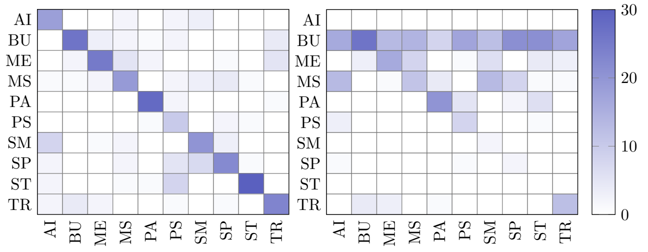
We also compare the performance of the proposed LoRA-BN method with other popular baseline methods in terms of average accuracy over current and previous domains in Table 2. Accuracy in the current domain and average forgetting over previous domains is also shown in Fig. 2. Results of FE compared to FT show that adapting the layers of the feature extractor to an incremental domain is better than freezing them. One can observe from Fig. 2a and 2b that higher forgetting of previous real domains $\mathcal{D}_1$ to $\mathcal{D}_3$ happens when the model starts learning the simulated domains, specifically $\mathcal{D}_4$ and $\mathcal{D}_5$ due to highly mismatched domains. The poor performance of FT in classifying the acoustic scenes from $\mathcal{D}_1$ after learning $\mathcal{D}_5$ can also be seen in Fig. 3b. This leads to a lower average accuracy for

[1] For S4-S6 the 3 hours of training data was not included in the official DCASE challenge train-test split, but is provided in the dataset.



(a) FE (b) FT (c) Disjoint

Figure 2: Accuracy at the current domain and average forgetting over previous domains of FE (a), FT (b) and disjoint (c) methods.



(a) (b)

Figure 3: Confusion matrices of a base model on domain $\mathcal{D}_1$ (a), FT on domain $\mathcal{D}_1$ after learning the simulated domain $D5$ corresponding to S2 (b). The 10 classes are, AI: airport, BU: bus, ME: metro, MS: metro station, PA: park, PS: public square, SM: shopping mall, SP: pedestrian street, ST: street with traffic, and TR: tram.

FE and FT for simulated domains $\mathcal{D}_4$ and $\mathcal{D}_5$, as seen in Table 2. However, FT uses all layers to adapt to the simulated devices after $\mathcal{D}_5$,and performs better overall after learning all domains.

The *disjoint* approach fine-tunes the base model trained on $\mathcal{D}_1$ to a current domain and performs well on real domains $\mathcal{D}_2$ and $\mathcal{D}_3$, maybe due to similar feature distributions. However, fine-tuning the base model directly to each simulated domain reduces the performance of the disjoint method as compared to FT, in which previous knowledge of the simulated domain is used to classify the acoustic scenes from the current domain.

The proposed LoRA-BN outperforms all other methods without forgetting any of the previously learned domains and its performance is close to the baseline joint which trains the model from the data of all previously seen domains. The number of LoRA parameters for each domain is 124434, which is only a 0.17% increase to the total parameters 75497930 of the base model. It shows that LoRA-BN is more suitable for practical scenarios because it only stores inexpensive LoRA parameters and running statistics.

We also compare the performance of LoRA-BN and other baseline systems in offline settings. Baseline systems suffer from overfitting and lead to decreased performance. However, LoRA-BN converges effectively over an increasing number of epochs with limited training data in incremental domains, as seen in Fig. 4. Further, we test the performance of all the methods by changing the order of the domains. We found that the devices S1 and S2 are more challenging to adapt in any order than other devices.

In comparison to the results of the DCASE Challenge 2020[2], the baseline achieves an average accuracy of 54.1%, being trained for 200 epochs on combined data of devices A-S3, with S4-S6 not included in the training. This result is aligned with the joint baseline in this paper, which achieves 54.7% using online training of all devices using the base model. DCASE baseline reports lower performance on simulated devices S1-S3, being trained offline, non-incrementally. Our proposed LoRA-BN achieves comparable results on S1-S3 when trained for 30 epochs, only on data of one device sequentially. However, our method follows a completely different learning procedure and is therefore not fully comparable with the DCASE baseline.

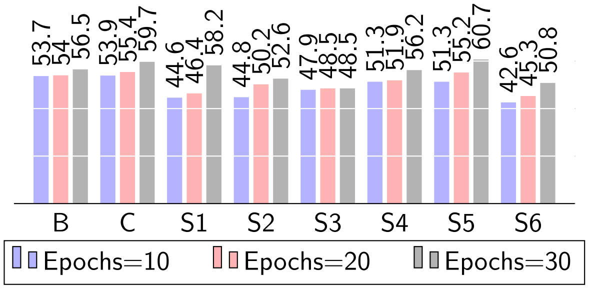


Figure 4: Accuracy of the LoRA-BN over increasing number of epochs.

4. CONCLUSION

In this paper, we propose a combination of LoRA parameters and running statistics of the BN layer for ODIL of acoustic scenes from different devices over time. Results show that highly mismatched simulated devices, especially starting devices S1 and S2 are more difficult to adapt by a model trained on real devices. ODIL-BN achieves poor performance on simulated devices and baselines severely forget acoustic scenes from previous real devices when these start learning simulated devices. The proposed LoRA-BN adapts effectively to the new domain and increases the performance of the base model by a large margin without forgetting acoustic scenes from any of the previously leaned devices. The performance of the LoRA-BN is further improved by increasing the number of iterations over the training data. LoRA-BN stores and uses inexpensive parameters and is more suitable for realistic applications. Future works include the development of a domain-agnostic approach that does not require device ID to classify acoustic scenes.

[2] https://dcase.community/challenge2020/task-acoustic-scene-classification#subtask-a-3

5. REFERENCES

[1] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural networks*, vol. 113, pp. 54–71, 2019.

[2] M. J. Mirza, M. Masana, H. Possegger, and H. Bischof, "An efficient domain-incremental learning approach to drive in all weather conditions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3001–3011.

[3] P. Garg, R. Saluja, V. N. Balasubramanian, C. Arora, A. Subramanian, and C. Jawahar, "Multi-domain incremental learning for semantic segmentation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 761–771.

[4] M. Mulimani and A. Mesaros, "Online domain-incremental learning approach to classify acoustic scenes in all locations," in *European Signal Processing Conference (EUSIPCO)*. IEEE, 2024.

[5] S. Gharib, K. Drossos, E. Cakir, D. Serdyuk, and T. Virtanen, "Unsupervised adversarial domain adaptation for acoustic scene classification," in *Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2018, pp. 138–142.

[6] K. Drossos, P. Magron, and T. Virtanen, "Unsupervised adversarial domain adaptation based on the wasserstein distance for acoustic scene classification," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2019, pp. 259–263.

[7] A. I. Mezza, E. A. Habets, M. Müller, and A. Sarti, "Unsupervised domain adaptation for acoustic scene classification using band-wise statistics matching," in *28th European Signal Processing Conference (EUSIPCO)*. IEEE, 2021, pp. 11–15.

[8] E. B. Zaken, Y. Goldberg, and S. Ravfogel, "Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2022, pp. 1–9.

[9] M. Wistuba, L. Balles, G. Zappella, *et al.*, "Continual learning with low rank adaptation," in *NeurIPS Workshop on Distribution Shifts: New Frontiers with Foundation Models*, 2023.

[10] S. Aleem, J. Dietlmeier, E. Arazo, and S. Little, "Convlora and adabn based domain adaptation via self-training," *arXiv preprint arXiv:2402.04964*, 2024.

[11] Z. Zhong, Z. Tang, T. He, H. Fang, and C. Yuan, "Convolution meets lora: Parameter efficient finetuning for segment anything model," in *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

[12] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.

[13] T. Heittola, A. Mesaros, and T. Virtanen, "Acoustic scene classification in dcase 2020 challenge: generalization across devices and low complexity solutions," in *Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2020, pp. 56–60.

[14] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "PANNs: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2880–2894, 2020.

[15] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," in *International Conference on Learning Representations (ICLR)*, 2017.

PRE-TRAINED MODELS, DATASETS, DATA AUGMENTATION FOR LANGUAGE-BASED AUDIO RETRIEVAL

Hokuto Munakata, Taichi Nishimura, Shota Nakada, Tatsuya Komatsu

LY Corporation, Japan

ABSTRACT

We investigate the impact of pre-trained models, datasets, and data augmentation on language-based audio retrieval. Despite the high interest in cross-modal retrieval and the introduction of various datasets, powerful encoders, and data augmentation techniques, it remains unclear which approaches are most effective for language-based audio retrieval. We focus on which should be selected to build a retrieval model. First, we investigate the performance gain by four audio encoders, PaSST, CAV-MAE, BEATs, and VAST, and three text encoders BERT, RoBERTa, and T5. Second, we prepare massive datasets of over 670k audio-text pairs including ClothoV2, AudioCaps, WavCaps, MACS, and Auto-ACD. Third, we investigate the combination of data augmentation methods to enhance the retrieval performance including mixup-contrast and text token masking. In addition, we also explore inference time augmentation by paraphrasing textual queries using Chat-GPT to achieve robust retrieval performance. Our final results achieve 39.79 points with a single model and 42.22 points with the ensemble models in the mean average precision among the top 10 results on the evaluation split of ClothoV2.

Index Terms— Language-based audio retrieval, Pre-trained model, Data augmentation,

1. INTRODUCTION

Language-based audio retrieval systems take a textual query as input and retrieve the corresponding audio from a database. The mainstream approach projects both audio and text data into a joint embedding space calculates their similarity, and ranks the audio based on this similarity [1] (Figure 1). To obtain this joint space, models are trained from audio-text pairs. Contrastive learning is a dominant training method, where positive audio-text pairs are given higher similarity scores and negative pairs lower scores [2].

This task shares common characteristics with text-to-image/video retrieval because both tasks involve processing language inputs and employ contrastive learning to train the dual encoders. In text-to-image/video retrieval, researchers have explored various encoders (e.g., CLIP [3], VideoBERT [4], BERT [5], RoBERTa [6]) trained on massive datasets with data augmentation methods (e.g., Mixco [7]). As with the visual domain, in the audio domain, the encoders, datasets, and data augmentation have been proposed [8, 9]. However, it remains unclear which approaches are most effective for audio retrieval. This motivates us to focus on which should be selected to train the retrieval model.

To address this, we investigate the impact of pre-trained models, datasets, and data augmentation on language-based audio retrieval. For pre-trained models, we use five audio encoders and three text encoders that have achieved state-of-the-art performance in downstream tasks. Specifically, we adopt PaSST [10], BEATs [11], CAV-MAE [12], and VAST [13] for the audio encoder and BERT [5], RoBERTa [6], and T5 [14]. For datasets, we prepare a massive dataset containing over 670k audio-text pairs. The dataset includes both manually annotated data such as ClothoV2 [1], AudioCaps [15], MACS [16], and WavCaps/Auto-ACD[17, 18] that utilize large language models (LLMs) to generate pseudo audio-text pairs. For data augmentation, we apply multiple data augmentation methods, mixup-contrast [7], and text token masking for further improvement of the retrieval performance. In addition, we also explore inference time augmentation by paraphrasing textual queries using Chat-GPT to achieve robust retrieval performance.

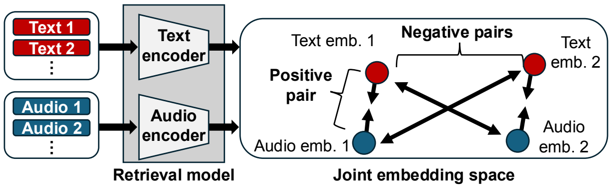


Figure 1: An overview of the conventional language-based audio retrieval system based on contrastive learning. Through contrastive learning, positive pairs of audio-text embeddings have similar values, while negative pairs have less similar values.

In our experiments, we conduct thorough comparative studies on encoders, datasets, data augmentation, and inference time augmentation. As a result, we provide the following three insights. First, in terms of encoders, VAST and RoBERTa yield the best performance. The performance of audio encoders aligned with the performance in the AudioSet classification task except for PaSST, which adopts patch-out, a regularization technique. Second, for the datasets, we observe that it is important to improve the text annotation quality, rather than increase the pseudo audio-text pairs generated from LLMs. Third, data augmentation approaches, both training and inference augmentation, contribute to the performance gain. As a result of combining these techniques, the model with PaSST and RoBERTa yields the best performance, achieving 39.79 mean average precision (mAP) on the ClothoV2 evaluation split. An ensemble of multiple models reaches 42.26 mAP.

2. MODEL OVERVIEW

As with recent cross-modal retrieval, our training approach is based on contrastive learning. The retrieval model has audio and text encoders to project the audio and text onto the joint embedding space. The input audio $\mathbf{X}^{(A)}$ and text $\mathbf{X}^{(T)}$ is projected onto D-dimensional joint space as $\mathbf{Z}^{(A)}$ and $\mathbf{Z}^{(T)}$ by audio/text encoders f_A and f_T as follows:

$$\mathbf{Z}^{(A)} = f_A(\mathbf{X}^{(A)}), \tag{1}$$

$$\mathbf{Z}^{(T)} = f_T(\mathbf{X}^{(T)}). \tag{2}$$

Based on $(\mathbf{Z}^{(A)}, \mathbf{Z}^{(T)})$, the model is trained to discriminate positive and negative from each pair of B audio and text samples based on InfoNCE [2] loss. Specifically, let i-th audio and j-th text be $\mathbf{Z}_i^{(A)}$ and $\mathbf{Z}_j^{(T)}$, where $i = j$ is positive and $i \neq j$ is negative. The loss is written as the cross-entropy loss with the softmax as follows:

$$\mathcal{L}_{\text{CE}}\left(\mathbf{Z}, \mathbf{Z}'_k, \mathbf{Z}'_{:}\right) = -\log \frac{\exp\left(S(\mathbf{Z}, \mathbf{Z}'_k)/\tau\right)}{\sum_{\mathbf{Z}' \in \mathbf{Z}'_{:}} \exp\left(S(\mathbf{Z}, \mathbf{Z}')/\tau\right)}, \quad (3)$$

$$\mathcal{L}_{\text{infoNCE}}^{A \to T} = \sum_i \mathcal{L}_{\text{CE}}\left(\mathbf{Z}_i^{(A)}, \mathbf{Z}_i^{(T)}, \mathbf{Z}_{:}^{(T)}\right), \quad (4)$$

$$\mathcal{L}_{\text{infoNCE}}^{T \to A} = \sum_j \mathcal{L}_{\text{CE}}\left(\mathbf{Z}_j^{(T)}, \mathbf{Z}_j^{(A)}, \mathbf{Z}_{:}^{(A)}\right), \quad (5)$$

$$\mathcal{L}_{\text{infoNCE}} = \mathcal{L}_{\text{infoNCE}}^{A \to T} + \mathcal{L}_{\text{infoNCE}}^{T \to A}, \quad (6)$$

where $\mathbf{Z}'_{:}$ is the set of $\mathbf{Z}'_1, ..., \mathbf{Z}'_B$, S is the cosine similarity, and τ is a trainable temperature parameter.

In the inference stage, the textual query and audio in the database are projected onto the joint embedding space, and their similarity is calculated to rank the audio. The audio ranked at position k in the database $\mathbf{X}_k^{(A)}$ is obtained by sorting the cosine similarities as follows:

$$\mathbf{X}_k^{(A)} = \underset{\mathbf{X}^{(A)} \in \mathbf{X}_{\text{DB}}^{(A)}}{\text{argmax}_k} \; S(f_A(\mathbf{X}^{(A)}), f_T(\mathbf{X}^{(T)})), \quad (7)$$

where argmax_k is the operation of extracting the k-th largest element and $\mathbf{X}_{\text{DB}}^{(A)}$ is the set of audio in the database.

3. AUDIO AND TEXT ENCODERS

3.1. Audio Encoder

The first focus of this study is encoders. For the audio side, we investigate four audio encoders: PaSST [10], BEATs [11], VAST [13], and CAV-MAE [12]. These models are variants of ASTs [19] that apply Vision Transformers [20] to audio spectra.

PaSST [10] enhances the AST by incorporating the patch-out technique that improves the generalization and accelerates the training by dropping the parts of the input sequence. Additionally, it employs distinct positional encodings for the time and frequency dimensions, leading to performance enhancements. We used the weights pre-trained on the AudioSet classification task. The stride size for the frequency and time was 16 and the patches were not overlapped. Only for this model, we apply patch-out [10] with 2 and 15 patches for the frequency and time directions during training.

CAV-MAE [12] extends the AST into an audio-visual model by integrating the outputs of AST and Vision Transformer [21]. This combined output is fed into a subsequent transformer that captures the interrelationships between audio and visual modalities through self-attention mechanisms. A multi-task loss that combines contrastive learning and masked autoencoder loss on both AudioSet and VGGSound datasets is used in the training. We used the scale++ model.

BEATs [11] introduces a discrete audio tokenizer to the AST framework, leveraging SSL. AST-based SSL model and the audio tokenizer are trained alternately in a repeated manner. Notably, BEATs demonstrated its high performance by being employed in the best system for the audio captioning task of the DCASE 2023 Challenge. We used the weights fine-tuned on the AudioSet classification task.

VAST [13] is a multi-modal model that integrates vision, audio, and texts into a unified framework using BEATs for the audio encoder. It is trained on the VAST-27M dataset, which includes 27 million video clips with vision-text or audio-text. The model trained with the dataset for various tasks such as retrieval, captioning, and question answering. VAST has demonstrated state-of-the-art performance on multiple cross-modality benchmarks. We used two different weights only pre-trained based on SSL and fine-tuned for the audio captioning task.

3.2. Text Encoder

For the text side, we investigate three text encoders: BERT [5], RoBERTa [6], T5 [14]. These models are based on Transformer architecture [22] trained with large-scale crawled text corpora. We use pre-trained weights of the large model of these encoders publicly available on HuggingFace.

BERT [5] is the bidirectional transformers encoder to improve understanding ability of the context of words in a sentence. This model is pre-trained based on masked language modeling (MLM) and next sentence prediction (NSP) with English Wikipedia and BookCorpus [23] containing over 3500 million words. We used the large model.

RoBERTa [6] is an optimized version of BERT pre-trained with diverse corpora of 160 GB. It removes NSP and focuses solely on MLM objectives. We used the large model.

T5 [14] is the transformer-based encoder and decoder architecture, which formulates a wide range of tasks such as sentence prediction. This model is trained based on SSL using multiple objective functions with C4 dataset [14], a web-crawled corpus of about 750 GB.

4. DATASET AND AUGMENTATION

4.1. Datasets

The second focus is the dataset. We prepare six datasets to investigate which one contributes to the retrieval performance: ClothoV2 [1], ClothoV2-GPT [24], MACS [16], AudioCaps [15], WavCaps [17], and Auto-ACD [18]. Note that all texts are preprocessed by removing punctuation and converting it to lowercase. In our evaluation, we use the ClothoV2 evaluation split and AudioCaps test split.

ClothoV2 [1] contains audio recordings ranging from ten to 30 seconds in length. The dataset is divided into training, validation, and test splits with 3840, 1045, and 1043 recordings, respectively. Each audio recording in the dataset is associated with five human-written captions containing eight to 12 words.

ClothoV2-GPT [24] is an augmented version of Clotho v2, where the original manually annotated text is expanded by five additional texts generated by OpenAI's GPT3.5-turbo. Five additional captions are generated by GPT based on the original audio's captions and keywords from metadata.

MACS [16] is extracted from the TAU Urban Acoustic Scenes 2019 and contains approximately 3,900 samples, each ten seconds long, totaling around 47 hours of audio. The captions are manually created, with roughly five captions per audio clip. The vocabulary size is 2803 words.

AudioCaps [15] is created by manually annotating a subset of The available subset of the dataset divided into training, validation, and test splits with 46163, 457, and 911 recordings, respectively. Most

of the clips are 10 seconds long. The captions are manually created, with one caption per audio clip. The vocabulary size is 5129 words.
WavCaps [17] includes samples from FreeSound, BBC Sound Effects, SoundBible, and AudioSetSL. It contains around 400k samples in total. The clip lengths vary from ten seconds to several minutes, with an average length of 67 seconds, totaling approximately 7500 hours of audio. Captions are automatically generated using GPT based on existing metadata (tags, etc.) and different prompts are used for each source dataset. Each audio clip has one caption, with a vocabulary size of 28721 words.
Auto-ACD [18] comprises samples from AudioSet and VGGSound [25]. We used the subset from VGGSound because it performs better on Clotho. It contains 180k samples generated from the YouTube video data of VGGSound. The text was generated by OpenAI's GPT leveraging existing tags and object recognition results from videos. Most clips are 10 seconds long, totaling approximately 500 hours of audio. Each audio clip has one caption, with a vocabulary size of 8157 words.

4.2. Training Data Augmentation

The third focus is the data augmentation. We use the following two approaches: Mix-up contrast (Mixco) [7] and text token masking.
Mixco [7] is a data augmentation method for contrastive learning. It was originally used for text-to-image retrieval and achieved significant performance gain. Mixco introduces the semi-positive pair, which is the pair of an image generated by mixing two images and their corresponding texts. To enable the model to learn better representations, the target labels for semi-positive pairs in the cross-entropy loss are set as soft labels rather than hard ones. To apply Mixco to language-based audio retrieval, we mix the i-th audio in the batch $\mathbf{X}_i^{(A)}$ and another audio $\mathbf{X}_{\phi(i)}^{(A)}$ in the waveform and transform it as follows:

$$\mathbf{X}_i^{(A')} = \lambda \mathbf{X}_i^{(A)} + (1-\lambda)\mathbf{X}_{\phi(i)}^{(A)}, \tag{8}$$

$$\mathbf{Z}_i^{(A')} = \mathsf{AudioEncoder}(\mathbf{X}_i^{(A')}), \tag{9}$$

where $\phi(i)$ is a randomly selected index for i and $\lambda \in (0,1)$ is a random variable sampled from the uniform distribution. From the embeddings of the mixtures, the additional loss of Mixco is obtained by the weighted sum of the infoNCE loss to discriminate semi-positive and negative pairs similar to Eq. (4) and Eq. (5) as follows:

$$\begin{aligned}\mathcal{L}_{\text{mixco}}^{A\to T} = \sum_i \lambda \Big\{ & \mathcal{L}_{\text{CE}}\left(\mathbf{Z}_i^{(A')}, \mathbf{Z}_i^{(T)}, \mathbf{Z}_:^{(T)}\right) \\ & +(1-\lambda)\,\mathcal{L}_{\text{CE}}\left(\mathbf{Z}_i^{(A')}, \mathbf{Z}_{\phi(i)}^{(T)}, \mathbf{Z}_:^{(T)}\right)\Big\},\end{aligned} \tag{10}$$

$$\begin{aligned}\mathcal{L}_{\text{mixco}}^{T\to A} = \sum_j \Big\{ & \lambda\mathcal{L}_{\text{CE}}\left(\mathbf{Z}_j^{(T)}, \mathbf{Z}_j^{(A')}, \mathbf{Z}_:^{(A')}\right) \\ & +(1-\lambda)\,\mathcal{L}_{\text{CE}}\left(\mathbf{Z}_{\phi(j)}^{(T)}, \mathbf{Z}_j^{(A')}, \mathbf{Z}_:^{(A')}\right)\Big\},\end{aligned} \tag{11}$$

$$\mathcal{L}_{\text{mixco}} = \mathcal{L}_{\text{mixco}}^{A\to T} + \mathcal{L}_{\text{mixco}}^{T\to A}. \tag{12}$$

We use the same temperature parameter for Eq. (3). In our experiment, we use the combination of the original info NCE loss and Mixco loss: $\mathcal{L} = \mathcal{L}_{\text{infoNCE}} + \mathcal{L}_{\text{mixco}}$.
Text token masking is a data augmentation method for the input text to mitigate overfitting. The text tokens are randomly replaced with `[MASK]` token for BERT and RoBERTa, and `<extra_id_0>` for T5. We set the replace probability to 15%.

Table 1: Performance by the audio and text encoders. The columns for mAP@10 represent the average and standard deviation achieved by the three models

ID	Audio encoder	Text encoder	mAP@10 ClothoV2	mAP@10 AudioCaps
A	PaSST	RoBERTa	39.77 ± .07	52.45 ± .32
B	CAV-MAE	RoBERTa	38.57 ± .77	51.52 ± .95
C	BEATs	RoBERTa	39.25 ± .14	54.70 ± .10
D	VAST (captioning)	RoBERTa	39.68 ± .09	**55.49 ± .06**
E	VAST (vanilla)	RoBERTa	**39.79 ± .14**	55.22 ± .31
F	PaSST	T5	36.06 ± .10	50.76 ± .23
G	PaSST	BERT	36.27 ± .20	49.03 ± .16

Table 2: Performance by the training dataset. The second column represents the number of audio-text pairs. The third column represents how the text data was created.

Training datasets	# of samples	mAP@10 ClothoV2	mAP@10 AudioCaps
1. ClothoV2	19k	27.30 ± .43	23.16 ± .28
2. ClothoV2-GPT	19k	27.36 ± .55	24.75 ± .51
3. AudioCaps	46k	23.59 ± .25	49.28 ± .18
4. WavCaps	401k	34.14 ± .38	44.22 ± .47
5. MACS	17k	8.30 ± .41	9.59 ± .42
6. Auto-ACD	185k	21.79 ± .26	28.82 ± .36
1 & 3 & 4	473k	38.40 ± .24	50.81 ± .36
2 & 3 & 4	473k	38.21 ± .09	51.05 ± .21
1 & 3 & 4 & 5	484k	38.80 ± .34	51.30 ± .07
1 & 3 & 4 & 6	651k	38.88 ± .33	51.92 ± .14
1 & 3 & 4 & 5 & 6	670k	**39.09 ± .43**	**52.18 ± .17**

4.3. Inference Time Augmentation

In addition to the training data augmentation, we also devise an inference time query augmentation method by paraphrasing textual queries using Chat-GPT to achieve robust retrieval performance. For example, a query of "A man walking who is blowing his nose hard and about to sneeze." is paraphrased to "A man walks while blowing his nose loudly" and "A man blows his nose hard as he walks." Since ClothoV2-GPT is generated only for the training split, we generated the same format dataset of the ClothoV2 evaluation split and AudioCaps test split using the same prompt of [24] except for not using the keywords. The text encoder projects the original and the additional queries and then the embeddings of each query are averaged.

5. EXPERIMENT

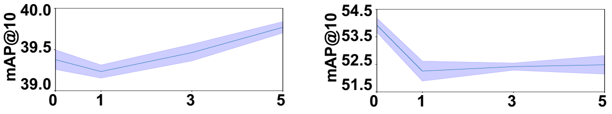
We conduct five experiments to confirm the effect of the encoders, datasets, training data augmentation, inference time augmentation, and the model ensemble.

5.1. Experimental Setting

The dimension of the joint embedding space is set to 1024. The number of training epochs and batch size are 15 and 128, respectively. The optimizer is AdamW [26]. The learning rate was changed by iterations using a cosine scheduler with 1 warm-up epoch and the maximum learning rate was 1×10^{-5}. The initial value of the temperature parameter τ used in Eq (3) was 0.02.

Table 3: Performance with and without data augmentation.

Mixco	Text token masking	mAP@10 ClothoV2	mAP@10 AudioCaps
-	-	39.09 ± .43	52.18 ± .17
✓	-	39.01 ± .27	49.25 ± .89
-	✓	39.33 ± .10	**52.83** ± .22
✓	✓	**39.77 ± .07**	52.45 ± .32



(a) ClothoV2 (b) AudioCaps

Figure 2: The relationship between the retrieval performance and the number of additional queries. The lines and areas represent the Average and standard deviation of mAP@10, respectively.

To avoid learning unexpected relationships between the audio and text caused by the difference among the datasets, we generate each batch from the same dataset. In the training, we conducted validation by 20% of each epoch and saved the model weight. After training, the weights of the models that achieved the top 10 in validation mAP@10 of ClothoV2 are averaged to form the final model weights. For the inference time augmentation, we generate five additional captions for ClothoV2 and the evaluation dataset of this challenge. The replacement probability of the original caption to the generated caption of ClothoV2-GPT is set to be 0.3 as with [24]. The preprocess and sampling rate of the audio follows the original implementation of each audio encoder. Audio clips are trimmed to 10 seconds if they are longer and padded if they are shorter. To mitigate performance variations due to initialization, we train all models three times. We performed the model ensemble by averaging the cosine similarity calculated by each model. In addition, to increase the diversity of the ensemble without additional training, two embeddings are obtained from a single model with or without inference time augmentation.

5.2. Results

Which encoders are the best? Table 1 shows the performance when changing the audio and text encoders. Note that in this experiment, we use all of the datasets, data augmentation, and inference time augmentation. We observe that RoBERTa achieves the best performance on both datasets and the two weights of VAST achieved the best performance on each dataset. In terms of the audio encoder, the performance in the AudioCaps is aligned with the performance in the AudioSet classification task. However, the performance in ClothoV2 is not aligned and PaSST is comparable to VAST. An important difference between PaSST and the other encoders is patch-out, encouraging PaSST to avoid overfitting. Among the text encoders, the performance of T5 is significantly worse than other models. This suggests that bi-directional text encoders (e.g., BERT and RoBERTa) are desirable for the language-based audio retrieval, rather than the uni-directional model (e.g., T5).

Which datasets have a significant impact on performance? Table 2 shows the performance when changing the training dataset. Note that in this experiment, we do not use data augmentation, and the audio/text encoders are PaSST and RoBERTa, respectively. When comparing models trained on ClothoV2 and ClothoV2-GPT, there was no significant performance difference, whether they were used as a single dataset (rows 1 and 2) or as subsets of multiple datasets (rows 7 and 8). The model trained only with MACS and Auto-ACD did not perform well (rows 5 and 6). In contrast, the model with WavCaps shows high performance for both evaluation datasets (row 4). When comparing the improvement of MACS and Auto-ACD (rows 8 and 9), despite the large difference in the number of samples, the difference in the performance was lower than 0.1 point for ClothoV2. When comparing Auto-ACD and WavCaps, the performance gap can be attributed to the fact that Auto-ACD does not implement the multiple filtering processes used by WavCaps. This suggests that acquiring high-quality text is crucial for training effective audio retrieval models.

Table 4: Performance of the ensembles of the multiple models

Model	mAP@10 ClothoV2	mAP@10 DCASE eval.
Ensemble of A, B, C, D, E	42.22	39.2
Our system of DCASE 2024	**42.26**	38.8
The best system of DCASE 2024	41.90	**41.6**
The best system of DCASE 2023	41.42	40.1

Which training data augmentation is the best? Third, we analyze the effect of the data augmentation and the summary is described in Table 3. In this experiment, we used all datasets, and the audio/text encoders are PaSST and RoBERTa. We obtain two findings. First, we separately conduct experiments on text token masking and Mixco and observe that text token masking slightly improves the performance yet Mixco does not. This may be because Mixco does not add new training text patterns, leading to the model's overfitting. Second, the combination of Mixco and text token masking significantly improves the performance. This result indicates that text token masking prevents the model from overfitting, enabling Mixco to be effective.

How many queries are necessary for inference time augmentation? Figure 2a and 2b show the performance change when varying the number of additional queries on ClothoV2 and AudioCaps. The results suggest that the number of additional queries depends on the datasets. In ClothoV2, five additional queries achieve the highest performance, whereas only one additional query has a negative impact. Based on these, we can say that the additional query supplements the missing information in the original query. In AudioCaps, this method degrades the performance even if we add five queries. This result implies that the text queries of AudioCaps already include enough keywords for the retrieval task.

Ensemble model performance. We measure the performance of the ensemble of A, B, C, D, and E that have different audio encoders, which is similar to our system of the DCASE 2024 Challenge. Although our model is comparable with the best system of the DCASE 2024 Challenge for ClothoV2, do not outperform it for the evaluation data of the DCASE Challenge. This result shows that we cannot avoid overfitting by merely using the large-scale dataset and encoders.

6. CONCLUSION

This report shows the impact of the audio and text encoders, datasets, and data augmentation methods. In our experiments, the single model achieved 39.79 points and the ensemble of the models achieved 42.22 points in mAP@10 on average for the ClothoV2 benchmark. Our future work includes the training strategy for large-scale datasets and pre-trained that can avoid overfitting.

7. REFERENCES

[1] K. Drossos, S. Lipping, and T. Virtanen, “Clotho: An audio captioning dataset,” in *Proc. ICASSP*, 2020, pp. 736–740.

[2] A. van den Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” *arXiv preprint arXiv:1807.03748*, 2018.

[3] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, “Learning transferable visual models from natural language supervision,” in *Proc. ICML*, 2021.

[4] C. Sun, A. Myers, C. Vondrick, K. Murphy, and C. Schmid, “VideoBERT: A joint model for video and language representation learning,” in *Proc. ICCV*, 2019.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proc. ACL*, 2019, pp. 4171–4186.

[6] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “RoBERTa: A robustly optimized bert pretraining approach,” *arXiv preprint arXiv:1907.11692*, 2019.

[7] S. Kim, G. Lee, S. Bae, and S.-Y. Yun, “Mixco: Mix-up contrastive learning for visual representation,” *arXiv preprint arXiv:2010.06300*, 2020.

[8] B. Elizalde, S. Deshmukh, and H. Wang, “Natural language supervision for general-purpose audio representations,” *arXiv preprint arXiv:2309.05767*, 2023.

[9] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, “Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation,” in *Proc. ICASSP*. IEEE, 2023, pp. 1–5.

[10] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, “Efficient training of audio transformers with patchout,” in *Proc. INTERSPEECH*, 2022.

[11] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, Z. Chen, W. Che, X. Yu, and F. Wei, “BEATs: Audio pre-training with acoustic tokenizers,” in *Proc. ICML*, 2023.

[12] Y. Gong, A. Rouditchenko, A. H. Liu, D. Harwath, L. Karlinsky, H. Kuehne, and J. R. Glass, “Contrastive audio-visual masked autoencoder,” in *Proc. ICLR*, 2022.

[13] S. Chen, H. Li, Q. Wang, Z. Zhao, M. Sun, X. Zhu, and J. Liu, “VAST: A vision-audio-subtitle-text omni-modality foundation model and dataset,” in *Proc. NeurIPS*, 2024.

[14] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, “Exploring the limits of transfer learning with a unified text-to-text transformer,” *J. Mach. Learn. Res.*, vol. 21, no. 140, pp. 1–67, 2020.

[15] C. D. Kim, B. Kim, H. Lee, and G. Kim, “AudioCaps: Generating captions for audios in the wild,” in *Proc. NAACL*, 2019.

[16] I. Martín-Morató, A. Mesaros, T. Heittola, T. Virtanen, M. Cobos, and F. J. Ferri, “Sound event envelope estimation in polyphonic mixtures,” in *Proc. ICASSP*. IEEE, 2019, pp. 935–939.

[17] X. Mei, C. Meng, H. Liu, Q. Kong, T. Ko, C. Zhao, M. D. Plumbley, Y. Zou, and W. Wang, “WavCaps: A chatgpt-assisted weakly-labelled audio captioning dataset for audio-language multimodal research,” *arXiv preprint arXiv:2303.17395*, 2023.

[18] L. Sun, X. Xu, M. Wu, and W. Xie, “A large-scale dataset for audio-language representation learning,” *arXiv preprint arXiv:2309.11500*, 2023.

[19] Y. Gong, Y.-A. Chung, and J. Glass, “AST: Audio spectrogram transformer,” in *Proc. INTERSPEECH*, 2021.

[20] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, “Training data-efficient image transformers & distillation through attention,” in *Proc. ICML*, 2021, pp. 10 347–10 357.

[21] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” in *Proc. ICLR*, 2021.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Proc. NeurIPS*, 2017.

[23] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, “Aligning books and movies: Towards story-like visual explanations by watching movies and reading books,” in *Proc. ICCV*, 2015.

[24] P. Primus, K. Koutini, and G. Widmer, “Advancing natural-language based audio retrieval with passt and large audio-caption data sets,” in *Proc. DCASE 2023 Workshop*, 2023.

[25] H. Chen, W. Xie, A. Vedaldi, and A. Zisserman, “VGGSound: A large-scale audio-visual dataset,” in *Proc. ICASSP*, 2020, pp. 721–725.

[26] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” *arXiv preprint arXiv:1711.05101*, 2017.

DATA-EFFICIENT ACOUSTIC SCENE CLASSIFICATION WITH PRE-TRAINING, BAYESIAN ENSEMBLE AVERAGING, AND EXTENSIVE AUGMENTATIONS

David Nadrchal[*], *Aida Rostamza*[*], *Patrick Schilcher*[*]

Johannes Kepler University Linz, Austria
{k12213656, k12237081, k12222369}@students.jku.at

ABSTRACT

The task of Acoustic Scene Classification (ASC) is to categorize short audio recordings into predefined scene classes. The DCASE community hosts an annual competition on ASC with a special focus on real-world problems such as recording device mismatches, low-complexity constraints, and limited labelled data availability. Solutions like Knowledge Distillation (KD) and task-specific data augmentations have proven effective in tackling these challenges, as demonstrated by their successful application in top-ranked systems. This paper contributes to the research on the real-world applicability of ASC systems by analyzing the effect of AudioSet pre-training on downstream training sets of different sizes. We study the impact of extensive data augmentation techniques, including Freq-MixStyle, device impulse response augmentation, FilterAugment, frequency masking, and time rolling on different training set sizes. Furthermore, the effectiveness of Bayesian Ensemble Averaging over traditional mean ensembling in KD is investigated. The results demonstrate that the proposed methods improve the performance over the DCASE baseline system substantially, with a particularly large gain on the smallest training set, lifting the accuracy by more than 7 percentage points on the development-test split. [1]

Index Terms— Acoustic Scene Classification, CP-Mobile, Knowledge Distillation, AudioSet pre-training, Bayesian Ensemble Averaging, Device Impulse Response augmentation, Freq-MixStyle, FilterAugment

1. INTRODUCTION

Acoustic Scene Classification (ASC) systems aim to categorize audio recordings into predefined scene classes. The Data-Efficient, Low-Complexity ASC task of the DCASE 2024 challenge [1] focuses on the real-world applicability of ASC systems by addressing three major problems, including recording device mismatch, low-complexity constraints, and limited training data availability. The task uses the TAU Urban Acoustic Scenes 2022 Mobile development dataset (TAU22) [2], consisting of 1-second audio recordings from 10 different scenes. The audio is recorded with three real devices, and six additional devices are simulated, with three of the simulated devices being only available in the test split, highlighting the importance of device generalization. The low-complexity constraints limit the model size to 128 kB and the computational complexity to 30 million multiply-accumulate operations (MACs), ensuring applicability on edge devices. The 2024 edition of this challenge addresses the real-world scenario of limited labelled data, requiring systems to maintain high accuracy with restricted training data across five scenarios: 5%, 10%, 25%, 50%, and 100% of the audio clips in the full training dataset. Systems must be trained exclusively on these subsets and explicitly allowed external resources, such as AudioSet [3].

This paper contributes to the research on the practical application of ASC systems by studying the effect of pre-training the student model on AudioSet. We examine the influence of various data augmentation techniques, such as Freq-MixStyle [4], device impulse response augmentation (DIR) [5], FilterAugment [6], frequency masking, and time rolling (shifting audio and wrapping segments that exceed the end back to the start) on different training set sizes. Additionally, the study evaluates the effectiveness of Bayesian Ensemble Averaging (BEA) compared to traditional mean ensembling in the context of Knowledge Distillation (KD). The proposed systems achieved the second rank in Task 1 of the DCASE 2024 challenge.

We review related work in Section 2, followed by a description of teacher and student architectures in Section 3. We then present the data-efficient training pipeline in Section 4. In Sections 5 and 6, we present the experimental setup and the results, respectively, and the paper is concluded in Section 7.

2. RELATED WORK

ASC Architectures: Convolutional Neural Networks (CNNs) have consistently proven to be leading models for low-complexity ASC [7, 8]. Restricting the receptive field of CNNs, known as Receptive Field Regularization [9, 10], has been shown to notably improve the generalization performance, with successful implementations in BC-ResNet [11] and CP-ResNet [9]. Inspired by efficient CNN architectures from the vision domain [12, 13], CP-Mobile (CPM) is a low-complexity, receptive-field regularized CNN for ASC, constructed of efficient inverted residual blocks. CPM achieved the top rank in the 2023 edition and a slightly simplified version of this architecture, excluding GRN [14], is used as the baseline system in the 2024 edition of this challenge. Recently, Audio Spectrogram Transformers (AST), such as the Patchout faSt Spectrogram Transformer (PaSST) [15], have shown state-of-the-art performance on various downstream tasks in the audio domain, including ASC.

Low-Complexity Techniques: Besides developing efficient architectures, several model compression techniques have been used in the context of ASC to meet the complexity requirements. In this regard, Pruning [16], Quantization [17], and, most importantly, KD [18] have become popular for further reducing system complexity. KD can be pointed out as the single most important technique for reducing the complexity of ASC systems, as it has been

[*] These authors contributed equally to this work

[1] Source Code: `https://github.com/SchilcherPatrick/DCASE24_Task1`

consistently used in top-ranked systems submitted to the challenge [19, 20, 21, 22, 8].

Recording Device Generalization: To tackle the device mismatch and generalization problem, various techniques have been explored, including Domain Adaptation [23, 24], training device translators [25], adjusting device sampling frequency [26], and normalizing data [4]. Among these, Freq-MixStyle [4] and DIR [5] augmentation techniques have proven to be particularly effective in boosting performance on unseen devices.

Data augmentation: Data augmentation is a widely used technique in ASC to improve model generalization and prevent overfitting, especially when dealing with small datasets and device mismatches. Some commonly used techniques include Mixup [27], SpecAugment [28], Freq-MixStyle [4], and DIR augmentation [5].

3. ARCHITECTURES

In this section, we present the teacher and student model architectures used in the KD-based training pipeline presented in Section 4.

3.1. TEACHER MODELS: PaSST, CP-ResNet, CP-Mobile

The Patchout faSt Spectrogram Transformer (PaSST) [15] is a self-attention-based AST model that excels in capturing global audio context and has achieved state-of-the-art performance on various downstream tasks in the audio domain [15]. By introducing the patchout mechanism for improved generalization and computational efficiency, PaSST has been shown to be an excellent teacher for low-complexity ASC models [29, 5].

CP-ResNet [9], a receptive-field regularized CNN, has been a successful model in previous ASC tasks [29, 9]. This fully convolutional architecture incrementally builds local features over a spatially restricted area. Receptive-field regularization has been shown to be important for improved generalization in ASC [9, 10].

CP-Mobile [30] (the student model in our setup; as described in the following section) is also included in the teacher ensemble, as described in Section 5.4.

3.2. STUDENT MODEL: CP-Mobile

Inspired by CP-ResNet, CPM is a factorized CNN architecture designed for low-complexity ASC, enhancing both representation capability and efficiency. The core innovation of CPM is the CPM block [22], a computationally efficient alternative to the classical convolutional layer that implements an inverted bottleneck block [12]. Each CPM block includes three factorized convolutional layers integrated with batch normalization and ReLU activation, targeting efficiency, and high representation capability, making CPM ideal for inference on edge devices.

4. DATA-EFFICIENT TRAINING PIPELINE

In this section, we introduce our proposed data-efficient training pipeline. We experiment with training the low-complexity student model, CPM, in three stages: Firstly, we pre-train CPM on AudioSet (Section 4.1), secondly, we train CPM on the respective train split of the TAU dataset (Section 4.2), and finally, we fine-tune CPM on the respective TAU split using KD (Section 4.3).

4.1. AudioSet Pre-Training

We hypothesize that pre-training the student model on a large general-purpose audio dataset, such as AudioSet [3], can reduce the need for extensive labelled data on downstream tasks. AudioSet contains over 2 million human-labeled 10-second sound clips across 527 distinct sound categories. This dataset provides a comprehensive resource for training and evaluating audio recognition models. General knowledge about acoustic events may improve performance on downstream tasks with limited training sets.

Following the training routine in [31], we train the CPM on AudioSet using KD from a large transformer ensemble of nine PaSST [15] models. Despite the high task complexity, the low-complexity network achieves a reasonable mean average precision performance of 0.194.

4.2. Pre-Training on Acoustic Scenes

Before distilling the knowledge from the teacher ensemble into the low-complexity student model, we train the student on the allowed TAU train split. The hypothesis is that pre-training on both AudioSet and TAU would provide a robust initialization by leveraging the diversity of AudioSet and the specific characteristics of TAU. It may also be beneficial for the learning process of the student to gain knowledge on acoustic scene data, before being exposed to the predictions of larger teacher ensembles.

4.3. Knowledge Distillation Fine-Tuning

KD compresses knowledge from a large, high-performing teacher model into a more compact student model while maintaining robust performance. Following [5], we train the student using both the soft targets (probability distributions over classes) from the teacher and hard labels (standard one-hot encoded labels). The overall loss function is defined as:

$$\text{Loss} = \lambda L_l(\delta(z_S), y) + (1-\lambda)\tau^2 L_{kd}(\delta(z_S/\tau), \delta(z_T/\tau)) \quad (1)$$

Here, the hard label loss (L_l) is the cross-entropy loss, and the distillation loss (L_{kd}) is the Kullback-Leibler divergence between the teacher's and student's soft targets. z_S and z_T are the logits of the student and teacher models, respectively, and y represents the hard labels. The temperature parameter (τ) controls the distribution sharpness, and the factor τ^2 is a scaling factor for the distillation loss. The contributions of both losses are balanced using a weight λ. This dual training approach allows the student model to capture both explicit label information and the generalized knowledge represented in the teacher's soft targets.

In this KD fine-tuning phase, we indirectly make use of AudioSet [3] a second time, by using KD with an AudioSet pre-trained teacher model, namely, the transformer PaSST [15].

4.3.1. Bayesian Ensemble Averaging

Ensembling teacher models is a common strategy to improve KD. By integrating diverse insights from the teachers, typically done by averaging their logits [5], this technique enhances the robustness and generalization of the student model.

Bayesian Ensemble Averaging (BEA) [32] extends simple averaging of logits by using a probabilistic framework. Inspired by BEA, we implemented a simplified interpretation without explicit distributional assumptions for model outputs. We used the average prediction of the teacher models as the expected prediction (μ_{tl}) and the logit-wise variance (σ_{tl}^2) across teacher models for each sample independently as a proxy for uncertainty. The aggregated prediction

(E_{tl}) combines the mean with a scaled variance to adjust the uncertainty impact based on the number of models (n_{tl}), ensuring proper moderation.

$$E_{tl} = \mu_{tl} + \frac{\sigma^2_{tl}}{n_{tl}} \tag{2}$$

5. EXPERIMENTAL SETUP

5.1. Audio Preprocessing

For all models, we downsample audio to a 32 kHz sampling rate. For the student, we compute Mel spectrograms using 256 frequency bins. The Short Time Fourier Transformation (STFT) is applied with a window size of 96 ms and a hop size of 16 ms. For the PaSST [15] teacher model, we follow its original AudioSet pre-training configuration and for CP-ResNet, the preprocessing remains the same as that of the CPM student except for the hop size being 24 ms.

5.2. Optimization

The student model is pre-trained on TAU using the Adam optimizer for 150 epochs. The training parameters include a weight decay of 0.0001, a learning rate of 0.005, and a warm-up phase of 2000 steps for the scheduler.

For the pre-training experiments, we shortened the KD training to 75 epochs and decreased the learning rate to 0.0025 as the model converged faster, due to prior knowledge of the domain. In other experiments, we applied the same hyperparameters for KD fine-tuning as those used in the student's pre-training on TAU.

As for CP-ResNet, the hyperparameters remain largely the same, with key differences being a learning rate of 0.001 and a weight decay of 0.001.

For PaSST, the learning rate and weight decay values are set to 0.00001 and 0.001, respectively. We use a patch out of 6 on the frequency dimension. The KD ensemble experiments including PaSST were trained using a learning rate of 0.0025

5.3. Data Augmentation

For all models, we use frequency masking of up to 48 frequency bins, time rolling of up to 0.1 seconds, and linear FilterAugment augmentation from 3 to 6 Mel bands in the range of -6 to 6 dB. Other augmentation hyperparameters fine-tuned for the different models are detailed in Table 1.

5.4. Knowledge Distillation

By default, we use an ensemble of one CPM and four CP-ResNet teachers for KD, with their predictions aggregated by BEA. The CP-ResNet teachers receive the same input as the student. In contrast, the PaSST teacher operates on its own spectrograms, which are independently subjected to frequency masking. Notably, Freq-MixStyle and FilterAugment are not applied to PaSST inputs. However, the time-domain augmentations, time-rolling and DIR, remain consistent for the PaSST teachers, the student, and the CP-ResNets.

We use temperature parameter $\tau = 2$ and Kullback-Leibler divergence with a high weight of $\lambda = 0.02$ as our loss function.

Training	**lr**	**DIR p**	**FMS p**	**FMS α**
CPM	0.005	0.6	0.6	0.4
CP-ResNet	0.001	0.6,0.7,0.8	0.6, 0.7, 0.8	0.3
PaSST	0.00001	0.6	0.4	0.4

Table 1: Hyper-parameters settings for different models. For the student, we use the same hyperparameters both for pre-training on TAU22 and for KD. For CP-ResNet, some hyperparameters differed among the teachers (indicated by multiple values in one cell), creating a diverse ensemble. FMS abbreviates Freq.-MixStyle [4], p stands for the probability that the respective augmentation is applied. α stands for the mixing alpha [4].

6. RESULTS

In this ablation study, we systematically add or remove specific system components to assess their impact on overall performance. This approach helps us understand each component's contribution to the model's accuracy and efficiency. The results reported are averages over three independent experiments.

Effect of Pre-training the Student Model: We evaluated the influence of pre-training the student model before KD by training it in three scenarios: without any pre-training (*Student with no pre-training*), pre-trained on the respective TAU split (*Student pre-trained TAU*), and pre-trained on both AudioSet and TAU (*Student pre-trained on AudioSet and TAU*). The results in Figure 1 and Table 2 with over 3% accuracy improvement on the smallest data split and over 1% on average across all splits, approve our hypothesis(4.2) and indicate that pre-training substantially enhances the model's performance. In contrast, the model trained without pre-training exhibits the lowest accuracy, underscoring the importance of pre-training. Pre-training on both AudioSet and TAU achieves the highest accuracy for the smallest datasets, while its impact is less pronounced in larger datasets, highlighting the effectiveness of AudioSet pre-training in handling minimal data in ASC.

Impact of Teacher Aggregation Methods in Knowledge Distillation: We investigated the impact of teacher aggregation methods in KD, comparing BEA (*KD-BEA*) and mean averaging for CP-ResNet and CP-Mobile teachers (*KD-Mean*). The results in Figure 2 and Table 3 suggest that BEA improves upon mean averaging across all TAU subsets when distilling knowledge to the student model pre-trained on the AudioSet and TAU split subset.

Contribution of including PaSST in the Teacher Ensemble: We evaluated the impact of including PaSST in the teacher ensemble alongside CP-ResNet and CP-Mobile teachers by examining three aggregation scenarios. The first scenario applied BEA to all teachers (*KD-BEA with PaSST*). The second used mean averaging for all teachers (*KD-Mean with PaSST*). The third scenario applied BEA for CP-ResNets and CP-Mobile and used mean averaging for the output of the BEA and the PaSST teacher logits (*KD-Mixed with PaSST*). The results in Figure 2 and Table 3 indicate that while both (*KD-BEA with PaSST*) and (*KD-Mean with PaSST*) perform similarly across all training subsets, the combination of BEA and mean aggregation (*KD-Mixed with PaSST*) methods demonstrates an overall superior performance. However, incorporating PaSST in all scenarios did not lead us to any performance improvements, likely due to resource limitations and the low number of training epochs for PaSST.

Effect of Various Data Augmentation Techniques on Student Model Generalization: We investigated the effect of various data augmentation techniques. As described in Figure 3 and Table 4, using FilterAugment instead of frequency masking (*Using FilterAugment, No Frequency Masking*) resulted in decreased performance across all training subsets. However, incorporating both FilterAug and frequency masking (*Using FilterAugment*), although not outperforming the proposed model system, demonstrated higher performance than alternating between the two. Removing frequency masking (*No Frequency Masking*) entirely led to higher performance in all but one subset compared to the default system. Freq.-MixStyle (*No Frequency Mixstyle*) showed improved performance in smaller subsets, while removing DIR (*No DIR*) caused an overall decrease in performance.

7. CONCLUSION

This paper introduces a data-efficient ASC system and examines various design choices concerning training sets of different sizes. We show that pre-training substantially boosts student model performance in a KD fine-tuning stage and can reduce the need for larger labelled datasets in downstream tasks. Pre-training on comprehensive datasets like AudioSet transfers general knowledge about acoustic events, enhancing model performance on downstream tasks with small training sets. Our simplified BEA can surpass mean aggregation in teacher ensembling. We explore the PaSST transformer's effectiveness for small training sets and assess various data augmentation techniques on model generalization. Our system improves performance over the DCASE 2024 baseline, achieving a 7 percentage point accuracy increase on the development-test split, especially with the smallest training set.

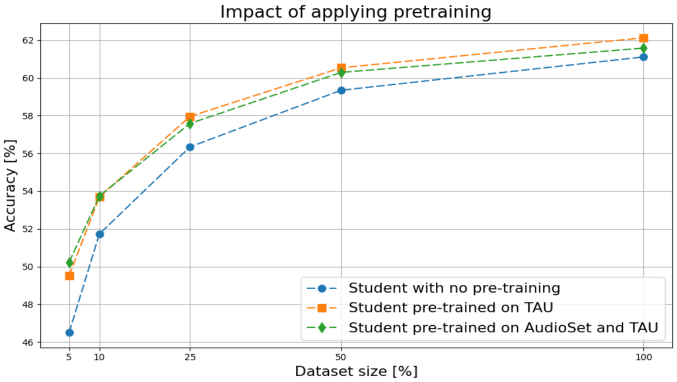


Figure 1: Effect of Pre-training the Student Model on AudioSet

Mode	5%	10%	25%	50%	100%	Avg.
-	46.49±.29	51.72±.12	56.34±.27	59.34±.48	61.11±.18	55.00
TAU	49.52±.15	53.68±.17	**57.96**±.31	**60.53**±.19	**62.13**±.27	**56.76**
AS, TAU	**50.22**±.10	**53.74**±.11	57.58±.20	60.29±.03	61.58±.11	56.68

Table 2: Impact of applying pre-training. AS and TAU indicate pre-training on AudioSet [3] and TAU, respectively, and **Avg.** denotes the average accuracy across all data splits.

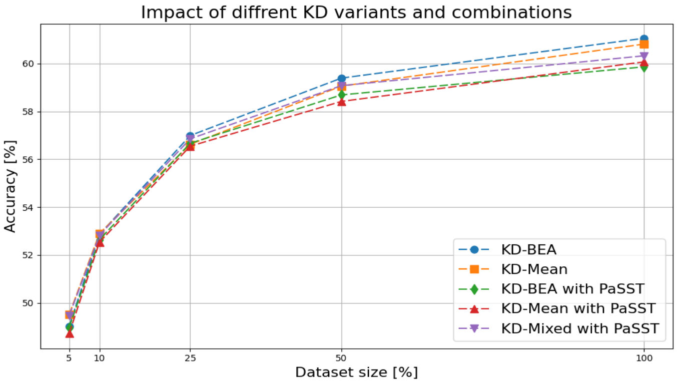


Figure 2: Impact of teacher aggregation methods

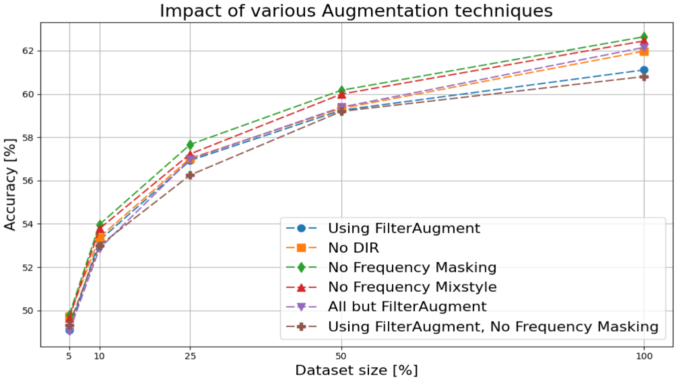


Figure 3: Effects of various data augmentation techniques

Mode	5%	10%	25%	50%	100%	Avg.
BEA	49.02±.48	52.85±.63	**56.99**±.36	**59.39**±.57	**61.05**±.78	**55.86**
Mean	49.12±.28	**52.89**±.45	56.62±.18	59.05±.39	60.81±.46	55.70
BEA+P	48.97±.28	52.64±.25	56.66±.14	58.68±.13	59.86±.35	55.36
Mean+P	48.72±.14	52.53±.03	56.54±.05	58.41±.13	60.06±.14	55.25
Mix+P	49.46±.28	52.82±.14	56.85±.21	59.07±.07	60.32±.17	55.71

Table 3: Impact of different KD variants and combinations. P indicates the inclusion of a PaSST model, and **Avg.** denotes the average accuracy across all data splits.

Method	5%	10%	25%	50%	100%	Avg.
All but FA	49.02±.48	52.85±.63	56.99±.36	59.39±.57	61.05±.78	55.86
+ FA	49.07±.24	53.23±.08	56.92±.07	59.25±.13	61.11±.10	55.92
- DIR	49.73±.41	53.37±.19	57.03±.12	59.33±.20	61.97±.37	56.29
- FM	**49.78**±.04	**53.96**±.10	**57.66**±.10	**60.17**±.16	**62.64**±.09	**56.84**
- FMS	49.64±.21	53.77±.22	57.23±.02	59.99±.10	62.45±.04	56.62
+ FA, - FM	49.32±.15	53.00±.13	56.25±.07	59.19±.06	60.80±.15	55.71

Table 4: Impact of different data augmentations. FA, FM, and FMS are abbreviate FilterAugment [6], frequency masking and Freq.-MixStyle [4], respectively, and **Avg.** denotes the average accuracy across all data splits.

8. REFERENCES

[1] DCASE Community, "Dcase 2024 challenge: Task - data-efficient low-complexity acoustic scene classification," https://dcase.community/challenge2024/task-data-efficient-low-complexity-acoustic-scene-classification, 2024, accessed: 2024-06-03.

[2] T. Heittola, A. Mesaros, and T. Virtanen, "TAU Urban Acoustic Scenes 2022 Mobile, Development dataset," 2022.

[3] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *ICASSP 2017*.

[4] B. Kim, S. Yang, J. Kim, H. Park, J. Lee, and S. Chang, "Domain generalization with relaxed instance frequency-wise normalization for multi-device acoustic scene classification," in *Interspeech 2022*, 2022.

[5] T. Morocutti, F. Schmid, K. Koutini, and G. Widmer, "Device-robust acoustic scene classification via impulse response augmentation," in *EUSIPCO 2023*, 2023.

[6] H. Nam, S. Kim, and Y. Park, "Filteraugment: An acoustic environmental data augmentation method," in *ICASSP 2022*, 2022.

[7] I. Martín-Morató, F. Paissan, A. Ancilotto, T. Heittola, A. Mesaros, E. Farella, A. Brutti, and T. Virtanen, "Low-complexity acoustic scene classification in DCASE 2022 challenge," in *DCASE 2022*.

[8] I. Martín-Morató, T. Heittola, A. Mesaros, and T. Virtanen, "Low-complexity acoustic scene classification for multi-device audio: Analysis of DCASE 2021 challenge systems," in *DCASE 2021*.

[9] K. Koutini, H. Eghbal-zadeh, and G. Widmer, "Receptive field regularization techniques for audio classification and tagging with deep convolutional neural networks," *IEEE ACM Trans. Audio Speech Lang. Process.*, 2021.

[10] K. Koutini, H. Eghbal-zadeh, M. Dorfer, and G. Widmer, "The receptive field as a regularizer in deep convolutional neural networks for acoustic scene classification," in *EUSIPCO 2019*.

[11] B. Kim, S. Yang, J. Kim, and S. Chang, "QTI submission to DCASE 2021: residual normalization for device-imbalanced acoustic scene classification with efficient design," *CoRR*, 2022.

[12] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *CVPR 2018*.

[13] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *ICML 2019*.

[14] S. Woo, S. Debnath, R. Hu, X. Chen, Z. Liu, I. S. Kweon, and S. Xie, "Convnext V2: co-designing and scaling convnets with masked autoencoders," in *CVPR 2023*.

[15] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, "Efficient training of audio transformers with patchout," in *Interspeech 2022*.

[16] J. Frankle, G. K. Dziugaite, D. M. Roy, and M. Carbin, "Pruning neural networks at initialization: Why are we missing the mark?" in *ICLR 2021*.

[17] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Quantized neural networks: Training neural networks with low precision weights and activations," *J. Mach. Learn. Res.*, 2017.

[18] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *CoRR*, 2015.

[19] H. Bing, H. Wen, C. Zhengyang, J. Anbai, C. Xie, F. Pingyi, L. Cheng, L. Zhiqiang, L. Jia, Z. Wei-Qiang, and Q. Yanmin, "Data-efficient acoustic scene classification via ensemble teachers distillation and pruning," DCASE2024 Challenge, Tech. Rep., 2024.

[20] D. Nadrchal, A. Rostamza, and P. Schilcher, "Data-efficient acoustic scene classification with pre-trained cp-mobile," DCASE2024 Challenge, Tech. Rep., 2024.

[21] Y.-F. Shao, P. Jiang, and W. Li, "Low-complexity acoustic scene classification with limited training data," DCASE2024 Challenge, Tech. Rep., 2024.

[22] F. Schmid, T. Morocutti, S. Masoudian, K. Koutini, and G. Widmer, "Cp-jku submission to dcase23: Efficient acoustic scene classification with cp-mobile," in *DCASE 2023*, 2023, pp. 161–165.

[23] K. Koutini, F. Henkel, H. Eghbal-zadeh, and G. Widmer, "CP-JKU submissions to DCASE'20: Low-complexity cross-device acoustic scene classification with RF-regularized CNNs," DCASE2020 Challenge, Tech. Rep., 2020.

[24] P. Primus, H. Eghbal-zadeh, D. Eitelsebner, K. Koutini, A. Arzt, and G. Widmer, "Exploiting parallel audio recordings to enforce device invariance in cnn-based acoustic scene classification," in *DCASE 2019*.

[25] B. Kim, S. Yang, J. Kim, and S. Chang, "QTI submission to DCASE 2021: Residual normalization for device-imbalanced acoustic scene classification with efficient design," DCASE2021 Challenge, Tech. Rep., 2021.

[26] J.-H. Lee, J.-H. Choi, P. M. Byun, and J.-H. Chang, "Hyu submission for the DCASE 2022: Efficient fine-tuning method using device-aware data-random-drop for device-imbalanced acoustic scene classification," DCASE2022 Challenge, Tech. Rep., 2022.

[27] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *ICLR 2018*.

[28] D. S. Park, W. Chan, Y. Zhang, C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," in *Interspeech 2019*, 2019.

[29] F. Schmid, S. Masoudian, K. Koutini, and G. Widmer, "Knowledge distillation from transformers for low-complexity acoustic scene classification," in *DCASE 2022*, 2022.

[30] F. Schmid, P. Primus, T. Heittola, A. Mesaros, I. Martín-Morató, K. Koutini, and G. Widmer, "Data-efficient low-complexity acoustic scene classification in the dcase 2024 challenge," 2024.

[31] F. Schmid, K. Koutini, and G. Widmer, "Efficient large-scale audio tagging via transformer-to-cnn knowledge distillation," in *ICASSP 2023*.

[32] J. Xu, S. Li, A. Deng, M. Xiong, J. Wu, J. Wu, S. Ding, and B. Hooi, "Probabilistic knowledge distillation of face ensembles," in *CVPR 2023*.

SELF TRAINING AND ENSEMBLING FREQUENCY DEPENDENT NETWORKS WITH COARSE PREDICTION POOLING AND SOUND EVENT BOUNDING BOXES

Hyeonuk Nam, Deokki Min, Seungdeok Choi, Inhan Choi, Yong-Hwa Park[*]

Korea Advanced Institute of Science and Technology, South Korea,
{frednam, minducky, haroldchoi6, ds5amk, yhpark}@kaist.ac.kr

ABSTRACT

To tackle sound event detection (SED), we propose *frequency dependent networks (FreDNets)*, which heavily leverage frequency-dependent methods. We apply frequency warping and FilterAugment, which are frequency-dependent data augmentation methods. The model architecture consists of 3 branches: audio teacher-student transformer (ATST) branch, BEATs branch and CNN branch including either partial dilated frequency dynamic convolution (PDFD conv) or squeeze-and-Excitation (SE) with time-frame frequency-wise SE (tfwSE). To train MAESTRO labels with coarse temporal resolution, we applied max pooling on prediction for the MAESTRO dataset. Using best ensemble model, we applied self training to obtain pseudo label from DESED weak set, unlabeled set and AudioSet. AudioSet pseudo labels, filtered to focus on high-confidence labels, are used to train on DESED dataset only. We used change-detection-based sound event bounding boxes (cSEBBs) as post processing for ensemble models on self training and submission models. The resulting FreDNet was ranked 2nd in DCASE 2024 Challenge Task 4.

Index Terms— frequency dynamic convolution, audio pretrained models, coarse prediction pooling, label filtering, sound event bounding boxes

1. INTRODUCTION

In this work, we address the problem of sound event detection (SED) with heterogeneous datasets, including Domestic Environment Sound Event Detection (DESED) and Multi-Annotator Estimated STROng labels (MAESTRO) [1, 2, 3]. Since SED is a very delicate task requiring classification with time localization, the difference between two datasets must be carefully addressed. While DESED uses hard labels with fine temporal resolution (base unit of one millisecond) and includes ten target sound events those occur in domestic environment, MAESTRO uses soft labels representing confidence with coarse temporal resolution (base unit of one second) and includes seventeen target sound events those occur on outdoor environments. There are only few target sound events overlapping. For the target sound events those do not overlap, target sound events from one dataset might exist in the other dataset but they are not explicitly labeled. This arouses the problem of potentially missing labels [1]. To tackle this problem, DCASE2024 Challenge Task 4 baseline is designed to train both datasets using single model architecture to output for 27 classes, while masking the classes from one dataset when training for the other dataset [1].

Our primary approach is to build strong classifier that works on both datasets. To achieve this, we applied two frequency-dependent data augmentations: frequency warping and FilterAugment [4, 5]. Then, we applied advanced variants of frequency dynamic convolution (FDY conv) to CNN branch of the baseline [6, 7, 8]. We also used squeeze and excitation (SE) with time-frame frequency wise SE (tfwSE) to CNN branch [9]. In addition to CNN and BEATs branch, we added audio teacher student transformer (ATST) branch to form three-branched models [4, 10, 11]. In order to match the granularity of strong prediction tailored for DESED to MAESTRO strong labels, we pooled strong predictions. Since frequency-dependent methods are heavily used, we call above network architecture as *Frequency Dependent Networks (FreDNets)*. We used change-detection-based sound event bounding boxes (cSEBBs) as post processing [12]. With ensemble of FreDNets post-processed by cSEBBs, we produced pseudo labels on AudioSets, and used them to train new FreDNets [13].

The main contributions of this paper are as follows:

1. Proposed *Frequency dependent networks (FreDNets)* heavily utilizes frequency-dependent methods to outperform the baseline by 15.1% without ensemble.
2. Proposed coarse prediction pooling successfully harmonizes the temporal resolution difference between the datasets.
3. Partial dilated frequency dynamic convolution (PDFD conv) is lighter than FDY conv or DFD conv providing various models, thus proven to be advantageous upon ensemble.

2. METHODS

2.1. Frequency-Dependent Data Augmentations

In addition to mixup applied in the baseline [1, 14], we added frequency warping and FilterAugment [4, 5]. The sequence of operation is as follows: mixup, frequency warping then FilterAugment. Frequency warping is random resize crop applied only along frequency dimension to zoom into frequency dimension with random proportion. As it also works as frequency shift, we did not apply frequency shift. Then, we applied linear type FilterAugment with dB range from -3 dB to +3 dB. This is narrower range compared to the setting in [5]. FilterAugment applies random weights over different frequency ranges to simulate different acoustic environments. Data augmentation is only applied to CNN branch as shown at the top of Fig. 1, because the other two branches are not trainable.

2.2. Frequency-Dependent CNN Methods

To further enhance the capacity of the network, CNN and RNN channels are doubled. Either variants of frequency dynamic convo-

[*]This work was supported by the Institute of Civil Military Technology Cooperation funded by the Defense Acquisition Program Administration and Ministry of Trade, Industry and Energy of Korean government under grant No. UM22409RD4, and Korea Research Institute of Ships and Ocean engineering a grant from Endowment Project of "Development of Open Platform Technologies for Smart Maritime Safety and Industries" funded by Ministry of Oceans and Fisheries(PES5230).

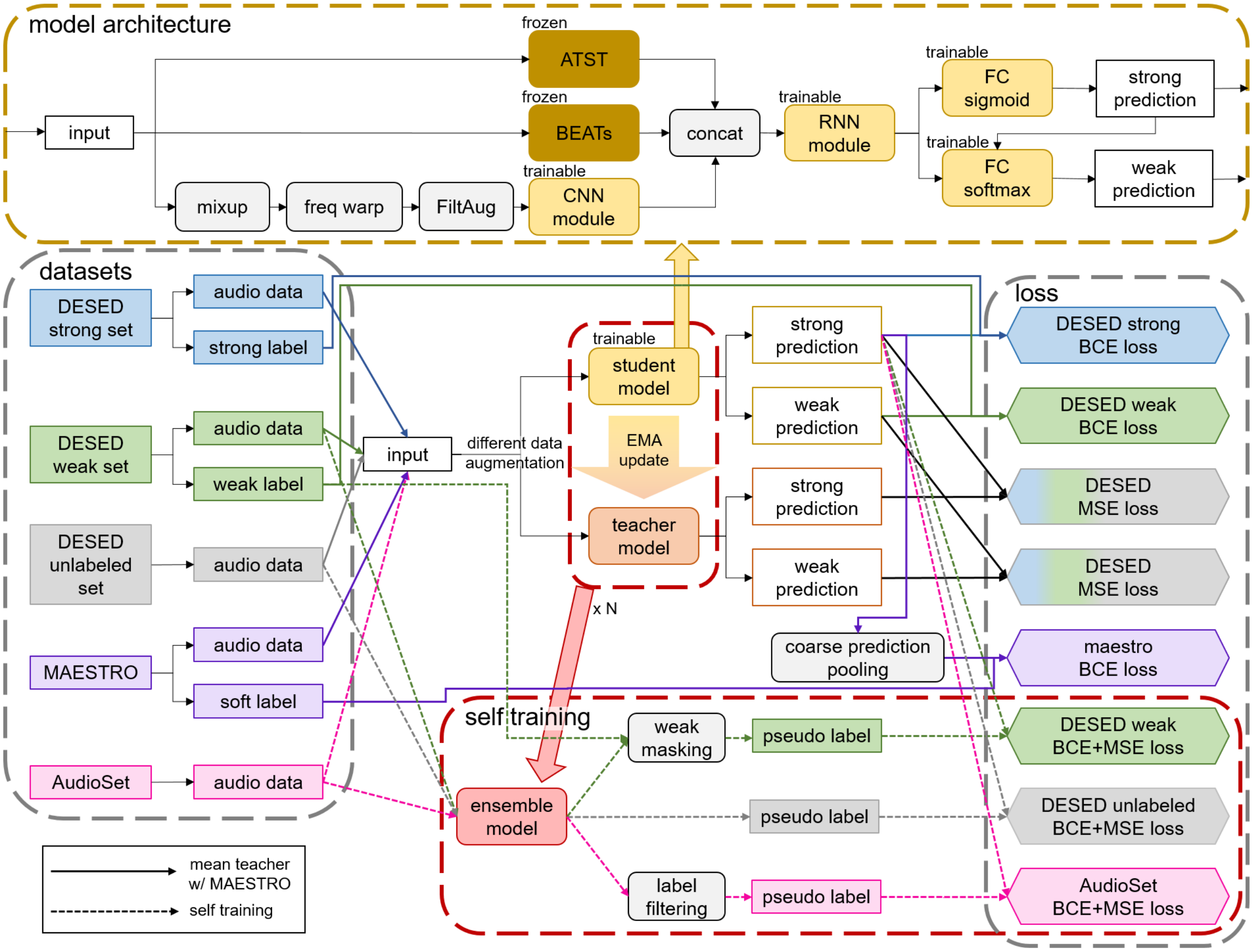


Figure 1: An illustration of framework for training and self training FreDNets.

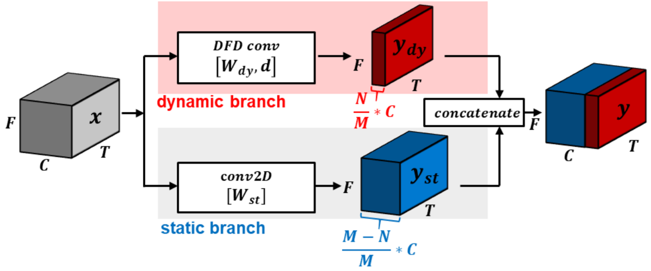


Figure 2: An illustration of partial dilated frequency dynamic convolution (PDFD conv). It involves a dynamic DFD conv branches and a static 2D convolution branch.

lution (FDY conv) or squeeze-and-excitation (SE) are used to make CNN modules leverage frequency-dependent attention methods.

FDY conv applies frequency-adaptive convolution kernel to release translational equivariance along frequency axis of time-frequency features [6]. To lighten FDY conv, we applied partial frequency dynamic convolution (PFD conv) with proportion of one over eight [8]. To expand and diversify the basis kernels, we applied dilated frequency dynamic convolution (DFD conv), which applies frequency-wise dilation to four basis kernels of PFD conv. We refer to this method as partial dilated frequency dynamic convolution (PDFD conv), which is illustrated in Fig. 2. Using different dilation sizes to PDFD conv resulted in various models which are advantageous on model ensemble [15]. While multi-dilated frequency dynamic convolution (MDFD conv) yields in the best performance, we used PDFD since it offers best cost-performance balance [8]. In addition to PDFD conv, we also used SE with time-frame frequency-wise SE (tfwSE) for model variety upon ensemble [9, 15].

2.3. Transformer-based Pre-trained Audio Models

In addition to CNN branch, two transformer-based pre-trained audio models are used: BEATs and ATST-frame. Frame-wise feature of BEATs and ATST-frame are used to optimally enhance SED which needs to give frame-wise predictions. Embeddings extracted for both methods are pooled into same frame size as output by CNN module output, then concatenated with the output from CNN module along channel dimension, and then processed by fully connected layers along channel dimension. Then the output is fed to RNN module. Note that since transformer-based Audio models divide mel spectrogram into patches and then apply positional encoding to the patches, they implicitly apply frequency-dependent processing. Thus two audio models can be regarded frequency-dependent methods as well. Fine tuning of ATST is not used in this work as it negatively affects MPAUC on MAESTRO [4].

2.4. Coarse Prediction Pooling

In order to address the different temporal resolution of DESED and MAESTRO, we applied coarse prediction pooling for MAESTRO data. While FreDNets' predictions have temporal resolution of 64ms per frame (156 frames for 10 seconds), MAESTRO label has temporal resolution of 1s per frame (10 frames for 10 seconds). To make fine predictions into coarse predictions, we apply max pooling on FreDNets' MAESTRO prediction. To be more specific, we zero-padded 2 frames before and after the prediction and max pooled with filter size and stride of 16. Although this is not precise pooling, this choice was made to quickly and simply implement the idea.

2.5. Sound Event Bounding Boxes

Polyphonic sound detection score (PSDS) applies various thresholds to the SED prediction to obtain threshold-independent evaluation values [16, 17]. However, as threshold differs, onset and offset of sound events also varies. To make onset and offset of sound events independent of the thresholds, sound event bounding boxes (SEBBs) are proposed to combine confidence values with very fine onset and offset values into representative confidence, onset and offset values [12]. In this work, change-detection-based SEBBs (cSEBBs) are used.

2.6. Self Training using AudioSet

To obtain pseudo labels on DESED weak set, DESED unlabeled set and AudioSet, we used ensemble of FreDNet using PDFD-CNN modules with varying dilation size sets, SE+tfwSE-CNN and PFD-CNN with varying seeds and then applied cSEBBs [15, 18]. As DESED weak set is given with weak labels, pseudo label for weak set is masked with given weak labels as in [19]. Since AudioSet has inconsistent label quality, we applied self training on whole dataset to obtain confidence from our ensemble FreDNet. For AudioSet, we filtered data files having pseudo label values (confidence) above 0.7 on 27 target events to focus on labels with high confidence. we discarded event labels with confidence value below 0.01 to reduce pseudo label metadata size, and removed the files of which events above 0.7 are only composed of subset of (speech, people talking, children voices) to reduce the data imbalance toward speech. The count of filtered AudioSet files is 153,977.

Upon use of AudioSet pseudo label, both soft label and hard label obtained by thresholding with 0.5 are used to train SED model. For mean square error (MSE) loss and binary cross entropy (BCE) loss are used for soft and hard labels respectively as shown in red dashed line box in Fig. 1. Only 10 target sound events for DESED are trained using filtered AudioSet as it degraded MPAUC when trained on MAESTRO target sound events, although it was meant to train on 17 target sound events in MAESTRO as well.

2.7. Ensemble

Ensemble model averaged predictions from various models. To maximize the effect of ensemble, we used different models including PFD-CRNN, PDFD-CRNN with different dilation size sets, SE+tfwSE-CRNN, and PFD-CRNN with different seeds. For each model setting, the student and teacher models with the best sum score (PSDS1+MPAUC) are selected for ensemble. The model combinations used for each ensemble setting is shown in Table 1. Ensemble 1 is used to extract pseudo labels from AudioSet. Ensemble 2 and 3 are used for DCASE Challenge submission. While PFD-CRNNs with different seeds are generally worse than models with seed of 42, models with different seeds do help enhancing ensemble performance.

Table 1: Components models of ensemble models. $1/8$ denotes that $1/8$ of PFD conv or PDFD conv output channel is from FDY conv or DFD conv. Sd, ds and st implies seed, dilation sizes and self training. For model names, CRNN is omitted for brevity.

ensemble	models
1	PFD(1/8), PFD(1/8, sd=2), PFD(1/8, sd=12), PFD(1/8, sd=16), PFD(1/8, sd=27), PFD(1/8, sd=34), PDFD(1/8, ds=1/1/2/2), PDFD(1/8, ds=1/1/3/3), PDFD(1/8, ds=2/2/3/3), PDFD(1/8, ds=1/1/2/3), PDFD(1/8, ds=1/2/2/3), PDFD(1/8, ds=1/2/3/3)
2	PFD(1/8), PFD(1/8, sd=16), PDFD(1/8, ds=1/1/2/2), PDFD(1/8, ds=1/1/3/3), PDFD(1/8, ds=1/1/2/3), PDFD(1/8, ds=1/2/2/3), PDFD(1/8, ds=1/2/3/3), st-PFD(1/8), st-PFD(1/8, sd=2), st-PFD(1/8, sd=12), st-SE+tfwSE, st-PDFD(1/8, ds=1/1/2/2), st-PDFD(1/8, ds=1/1/2/3), st-PDFD(1/8, ds=1/2/2/3), st-PDFD(1/8, ds=1/2/3/3)
3	PFD(1/8), PFD(1/8, sd=16), SE+tfwSE, PDFD(1/8, ds=1/1/2/2), PDFD(1/8, ds=1/1/3/3), PDFD(1/8, ds=1/1/2/3), PDFD(1/8, ds=1/2/2/3), PDFD(1/8, ds=1/2/3/3), st-PFD(1/8), st-PFD(1/8, sd=2), st-PFD(1/8, sd=12), st-PFD(1/8, sd=27), st-SE+tfwSE, st-PDFD(1/8, ds=1/1/2/2), st-PDFD(1/8, ds=1/1/2/3), st-PDFD(1/8, ds=1/2/2/3), st-PDFD(1/8, ds=1/2/3/3),

3. EXPERIMENTAL SETTINGS

3.1. Implementation Details

DESED and MAESTRO data are processed to be 10 seconds clip with 16kHz sampling rate [1, 3, 20]. Mel spectrogram is used for input feature. The network is composed of three-branched ATST-BEATs-CNN modules which are then fed to RNN module and Fully Connected layers as shown in Fig. 1. The Mean Teacher method is employed to train FreDNets using the DESED unlabeled set [20, 21]. Binary cross entropy (BCE) loss is used to train strong prediction for DESED strong set and its strong label, weak prediction for DESED weak set and its weak label, and strong prediction of MAESTRO and its soft label. Note that strong prediction goes through coarse label prediction before the loss function to match the granularity of prediction and label. For consistency loss for strong and weak predictions of DESED sets, mean square error (MSE) loss is used. For pseudo labels for DESED weakly labeled set, unlabeled set and AudioSet, both BCE and MSE losses are used. Default seed is set to 42. GPU used for training is NVIDIA RTX A6000. For post-processing, we use either cSEBBs or a median filter as reported in Table 2. The median filter refers to class-independent 7-frames-sized median filter.

3.2. Evaluation Metrics

True PSDS1 was used to evaluate SED performance on DESED [16, 17]. While previous DCASE challenge task 4 used two types of PSDS (PSDS1 favoring time localization and PSDS2 favoring accurate classification), only PSDS1 is used in this year as PSDS2

Table 2: Performance of FreDNets.

models	pre-trained models	post-processing	self training	PSDS1	MPAUC	sum	# submission
Baseline [1]	BEATs	median filter	-	0.520	0.637	1.157	-
PFD-CRNN(1/8)	ATST + BEATs	median filter	-	0.516	0.775	1.293	-
PFD-CRNN(1/8, sd=2)	ATST + BEATs	median filter	-	0.502	0.766	1.268	-
PFD-CRNN(1/8, sd=12)	ATST + BEATs	median filter	-	0.514	0.765	1.279	-
PFD-CRNN(1/8, sd=16)	ATST + BEATs	median filter	-	0.514	0.772	1.286	-
PFD-CRNN(1/8, sd=27)	ATST + BEATs	median filter	-	0.514	0.763	1.277	-
PFD-CRNN(1/8, sd=34)	ATST + BEATs	median filter	-	0.508	0.769	1.276	-
PDFD-CRNN(1/8, 1122)	ATST + BEATs	median filter	-	0.519	0.773	1.292	-
PDFD-CRNN(1/8, 1133)	ATST + BEATs	median filter	-	0.523	0.767	1.290	-
PDFD-CRNN(1/8, 2233)	ATST + BEATs	median filter	-	0.515	0.772	1.287	-
PDFD-CRNN(1/8, 1123)	ATST + BEATs	median filter	-	0.518	**0.776**	1.294	-
PDFD-CRNN(1/8, 1223)	ATST + BEATs	median filter	-	**0.526**	0.772	**1.298**	-
PDFD-CRNN(1/8, 1233)	ATST + BEATs	median filter	-	0.518	0.774	1.292	-
SE+tfwSE-CRNN	ATST + BEATs	median filter	-	0.507	0.773	1.280	-
Ensemble 1	ATST + BEATs	median filter	-	**0.527**	**0.790**	**1.317**	-
Ensemble 1	ATST + BEATs	cSEBBs	-	**0.577**	**0.790**	**1.367**	-
PFD-CRNN(1/8)	ATST + BEATs	median filter	True	**0.539**	0.773	**1.312**	-
PFD-CRNN(1/8, sd=2)	ATST + BEATs	median filter	True	0.534	0.766	1.300	-
PFD-CRNN(1/8, sd=12)	ATST + BEATs	median filter	True	0.534	0.753	1.287	-
PFD-CRNN(1/8, sd=27)	ATST + BEATs	median filter	True	0.531	0.750	1.287	-
PDFD-CRNN(1/8, 1122)	ATST + BEATs	median filter	True	0.530	0.774	1.304	-
PDFD-CRNN(1/8, 1133)	ATST + BEATs	median filter	True	0.535	0.761	1.296	-
PDFD-CRNN(1/8, 1123)	ATST + BEATs	median filter	True	0.537	**0.775**	**1.312**	-
PDFD-CRNN(1/8, 1223)	ATST + BEATs	median filter	True	0.533	0.772	1.305	-
PDFD-CRNN(1/8, 1233)	ATST + BEATs	median filter	True	0.532	0.772	1.304	-
SE+tfwSE-CRNN	ATST + BEATs	median filter	True	0.525	0.767	1.292	-
PFD-CRNN(1/8)	ATST + BEATs	cSEBBs	True	0.551	0.773	1.324	1
PDFD-CRNN(1/8, 1123)	ATST + BEATs	cSEBBs	True	**0.557**	**0.775**	**1.332**	2
Ensemble 2	ATST + BEATs	median filter	True	**0.537**	0.788	**1.325**	-
Ensemble 3	ATST + BEATs	median filter	True	0.536	**0.789**	**1.325**	-
Ensemble 2	ATST + BEATs	cSEBBs	True	**0.575**	0.788	**1.363**	3
Ensemble 3	ATST + BEATs	cSEBBs	True	0.574	**0.789**	**1.363**	4

is rather an audio tagging metric [12, 19]. For MAESTRO performance evaluation, MPAUC is used [1]. We optimized the model based on average score of PSDS1 + MPAUC on 4 independent training runs. The scores reported in the table are from the models with best sum scores among 4 independent training runs within each model setting.

4. RESULTS

The results are summarized in Table 2, highlighting the performance improvements achieved by our proposed methods. The PSDS and MPAUC values are obtained on real validation sets of DESED and MAESTRO respectively. As shown in the results, PFD-CRNN and PDFD-CRNNs do not significantly vary in their performance. However, as their roles differ from each other, ensembling differently dilated PDFD-CRNNs results in decent performance. Likewise, although slightly worse than PDFD-CRNNs, SE-tfwSE-CRNN and PFD-CRNNs with different seeds do help for ensemble. From the results, it could be inferred that use of FreDNet including frequency-wise data augmentation, PDFD conv, BEATs, ATST-frame, coarse prediction pooling enhances MPAUC by large margine while PSDS is not significantly improved. Rather, use of cSEBBs and self training improves PSDS significantly. Final best score without ensemble model outperforms the baseline by 15.1% and best score with ensemble outperforms the baseline by 18.2%. While ensemble 1 model slightly outperforms ensemble 2 and 3 those outperformed the baseline by 17.8%, submission was made with latter two as they contain self-trained models thus are expected to retain better generalization capability.

5. CONCLUSION

In this study, we presented Frequency Dependent Networks (FreDNet) for SED. FreDNet leverages frequency-dependent data augmentation techniques, frequency warping and FilterAugment, and incorporates advanced neural network architectures such as frequency dependent CNNs and transformer-based pre-trained models. Experiments show that the proposed FreDNet architecture, when combined with PDFD conv, SE, and coarse prediction pooling, significantly improves SED performance especially on MPAUC. The use of cSEBBs further enhances performance by refining onset and offset predictions on PSDS. The ensemble models, integrating various FreDNet settings, achieved substantial performance gains over the baseline, with the best ensemble model outperforming the baseline by 18.2%. Our approach shows promise for robust SED in diverse environments, highlighting the effectiveness of frequency-dependent methods and the importance of ensemble strategies in improving model performance. The model described in this work was ranked 2nd in DCASE 2024 Challenge Task 4.

6. REFERENCES

[1] S. Cornell, J. Ebbers, C. Douwes, I. Martín-Morató, M. Harju, A. Mesaros, and R. Serizel, "Dcase 2024 task 4: Sound event detection with heterogeneous data and missing labels," *arXiv preprint arXiv:2406.08056*, 2024.

[2] N. Turpault, R. Serizel, A. Parag Shah, and J. Salamon, "Sound event detection in domestic environments with weakly labeled data and soundscape synthesis," in *Workshop on Detection and Classification of Acoustic Scenes and Events*, 2019.

[3] I. Martín-Morató and A. Mesaros, "Strong labeling of sound events using crowdsourced weak labels and annotator competence estimation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

[4] N. Shao, X. Li, and X. Li, "Fine-tune the pretrained atst model for sound event detection," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024.

[5] H. Nam, S.-H. Kim, and Y.-H. Park, "Filteraugment: An acoustic environmental data augmentation method," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[6] H. Nam, S.-H. Kim, B.-Y. Ko, and Y.-H. Park, "Frequency dynamic convolution: Frequency-adaptive pattern recognition for sound event detection," in *Proc. Interspeech*, 2022.

[7] H. Nam, S.-H. Kim, D. Min, J. Lee, and Y.-H. Park, "Diversifying and expanding frequency-adaptive convolution kernels for sound event detection," *arXiv preprint arXiv:2406.05341*, 2024.

[8] H. Nam and Y.-H. Park, "Pushing the limit of sound event detection with multi-dilated frequency dynamic convolution," *arXiv preprint arXiv:2406.13312*, 2024.

[9] H. Nam, S.-H. Kim, D. Min, and Y.-H. Park, "Frequency & channel attention for computationally efficient sound event detection," in *Workshop on Detection and Classification of Acoustic Scenes and Events*, 2023.

[10] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, Z. Chen, W. Che, X. Yu, and F. Wei, "Beats: Audio pre-training with acoustic tokenizers," in *International Conference on Machine Learning*, 2023.

[11] X. LI and X. Li, "Atst: Audio representation learning with teacher-student transformer," in *Proc. Interspeech*, 2022.

[12] J. Ebbers, F. G. Germain, G. Wichern, and J. L. Roux, "Sound event bounding boxes," *arXiv preprint arXiv:2406.04212*, 2024.

[13] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[14] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations*, 2018.

[15] H. Nam, S.-H. Kim, D. Min, B.-Y. Ko, S.-D. Choi, and Y.-H. Park, "Frequency dependent sound event detection for dcase 2022 challenge task 4," DCASE2022 Challenge, Tech. Rep., 2022.

[16] Ç. Bilen, G. Ferroni, F. Tuveri, J. Azcarreta, and S. Krstulović, "A framework for the robust evaluation of sound event detection," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 61–65.

[17] J. Ebbers, R. Haeb-Umbach, and R. Serizel, "Threshold independent evaluation of sound event detection scores," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[18] J. W. Kim, S. W. Son, Y. Song, H. K. Kim, I. H. Song, and J. E. Lim, "Semi-supervised learning-based sound event detection using frequency dynamic convolution with large kernel attention for DCASE challenge 2023 task 4," DCASE2023 Challenge, Tech. Rep., 2023.

[19] H. Nam, B.-Y. Ko, G.-T. Lee, S.-H. Kim, W.-H. Jung, S.-M. Choi, and Y.-H. Park, "Heavily augmented sound event detection utilizing weak predictions," DCASE2021 Challenge, Tech. Rep., 2021.

[20] N. Turpault. Dcase2021 task4 baseline. GitHub. Available: https://github.com/DCASE-REPO/DESED_task. [Online]. Available: https://github.com/DCASE-REPO/DESED_task

[21] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

IMPROVING DOMAIN GENERALISATION WITH DIVERSITY-BASED SAMPLING

Andrea Napoli, Paul White

Institute of Sound and Vibration Research
University of Southampton, UK
{an1g18, P.R.White}@soton.ac.uk

ABSTRACT

Domain shifts are a major obstacle to the deployment of automated bioacoustic monitoring tools to new recording environments or habitats. Invariance regularisation is one approach for dealing with these shifts, in which the feature distributions of data from different domains are encouraged to match (by minimising some measure of statistical distance). However, in a deep learning setup, the statistical distance is only computed over small minibatches of data at a time. Inevitably, small samples have poor representation of their underlying distributions, resulting in extremely noisy distance estimates. In this paper, we propose that promoting wider distribution coverage, by inducing diversity in each sampled minibatch, would improve these estimates, and hence the generalisation power of the trained model. We describe two options for diversity-based data samplers, based on the k-determinantal point process (k-DPP) and the k-means++ algorithm, which can function as drop-in replacements for a standard random sampler. We then test these on a domain shift task based on humpback whale detection, where we find both options improve the performance of two invariance regularisation algorithms, as well as standard training via empirical risk minimisation (ERM).

Index Terms— domain shift, bioacoustics, invariance regularisation, determinantal point process

1. INTRODUCTION

Machine learning methods often underperform on data lying outside the training distribution. The sensitivity to distributional shifts (also called domain shifts) is currently a severe limitation to the widespread deployment of AI to real-world problems. This issue manifests widely, and significant research effort has already been invested towards achieving better out of distribution (OOD) generalisation [1, 2].

Invariance regularisation (also referred to as invariant feature learning or distribution alignment) is possibly the dominant approach in this field. Given meta-data which groups samples according to certain characteristics or contexts (referred to as *domains*), the technique aims to learn feature representations which are invariant to these characteristics, in the hope that this increases the generalisation power of the learned model. If unlabelled data from the test domain is included, this technique is referred to as unsupervised domain adaptation (UDA).

In practice, invariance regularisation has manifested in two main ways:

1) as an additional (differentiable) term in the loss function describing the statistical distance between data batches from different domains, which is minimised alongside the standard objective of empirical risk (ERM). Such distance measures include squared differences in second order statistics (mean and covariance) [3] or the maximum mean discrepancy (MMD) [4].

2) via domain-adversarial training [5, 6], in which a discriminator network is trained to predict which domain the features belong to, and the feature extractor is tuned to maximise discriminator error, alongside the loss on the main task. Depending on the exact formulation, it has been shown that adversarial networks minimise the Jensen-Shannon divergence or Wasserstein distance between domains [7].

Although these techniques employ clever tricks that circumvent having to estimate the distributions directly, the distance measures still require the samples to properly capture their underlying distributions. In this context, where the feature space is high-dimensional and the sample sizes are small, this becomes all the more important. In practice, and possibly because of this, invariance regularisation has frequently been found to have a negligible or even negative impact on training compared to vanilla ERM [1, 2, 8–10].

In this paper, we propose that minibatches that better cover the support of their underlying distribution would give higher quality distance estimates, and thus increase the effectiveness of invariance regularisation methods. We propose to achieve this by inducing diversity in each sampled minibatch – corresponding to the datapoints being "spread out" (pairwise dissimilar) in the learned model's feature space.

We note that this approach can also be interpreted as a generalisation of class-balancing. Inevitably, complex real-world acoustic scenes have a far richer ontology than the fixed set of class labels provided for the specific learning task (which may only be binary). So, this method can be motivated by the same logic as why classes are normally balanced prior to training: to ensure equal representation of all sound events.

Thus, the requirement is for a fast, scalable sampler which can stochastically draw independent, diverse minibatches of fixed cardinality from the corpus. It should also be possible to weight each instance to bias its selection probability based on prior knowledge, e.g., the label distribution – although, note, we are not interested in explicitly class-balancing the data, as doing so is at odds with the objective of diversity: some classes (e.g., "not a humpback whale") may have far greater variety than others. Also note that, given feature-label continuity, inducing diversity does tend to implicitly class-balance the data anyway [11].

We identify two options which satisfy these desiderata: the k-determinantal point process (k-DPP) and the k-means++ algorithm, which are discussed next.

This work was supported by grants from BAE Systems and the Engineering and Physical Sciences Research Council.

1.1. Determinantal point process (DPP)

Given a set of feature embeddings $\mathcal{X} = \{x_1, \dots, x_n\}$, $x_i \in \mathbb{R}^d$, a point process on $\mathcal{X}$ is a probability measure over "point configurations" (i.e., subsets) of $\mathcal{X}$. Sampling a point process is thus equivalent to randomly drawing a subset of $\mathcal{X}$. For a determinantal point process (DPP) [12], the probability of drawing subset $\mathcal{A}$ is proportional to the determinant of a likelihood kernel $L_{\mathcal{A}}$ describing pairwise similarities between its elements. Specifically:

$$\mathbb{P}[\mathcal{S}] = \frac{\det L_{\mathcal{A}}}{\det [I + L]}, \qquad \forall \mathcal{S} \subseteq \mathcal{X}, \tag{1}$$

where $L \in \mathbb{R}^{n \times n}$ is the kernel over all $\mathcal{X}$. When the DPP is conditioned to a fixed cardinality $|\mathcal{A}| = k \leq \text{rank}(L)$, this is known as a k-DPP.

As a result of (1), large off-diagonal entries in L imply low probability of co-occurrence in $\mathcal{A}$; this makes DPPs a common tool for inducing diversity. Previous use cases of DPPs in bioacoustics include to facilitate the exploration of large corpuses [13], and to implicitly class-balance unlabelled data for semi-supervised learning and unsupervised domain adaptation [11].

To apply weights $w = [w_1, \dots, w_n]^T$ to each instance, we can define L based on a similarity matrix S, with each element weighted by the corresponding pair of weights: $L_{ij} = \sqrt{w_i w_j} S_{ij}$.

Thus, the k-DPP is now restricted to $k \leq \text{rank}(S)$. An appropriate choice of similarity measure should ensure that S is full rank (that is, the kernel should be *strictly* positive-definite), so as not to limit the minibatch size we can use. For example, the commonly-used linear kernel $S_{ij} = x_i^T x_j$ results in *at best* $k \leq d$, but this could well be lower if the features are not all linearly independent. Therefore, in this paper, we propose to use the radial basis function (RBF) kernel instead. Specifically, adopting a common heuristic for the bandwidth parameter γ [4], we use an RBF mixture kernel defined by

$$S_{ij} = \sum_{\gamma \in \mathcal{G}} e^{-\gamma \|x_i - x_j\|^2} \tag{2}$$

with $\mathcal{G} = \{0.001, 0.01, 0.1, 1, 10\}$. That the RBF is always strictly positive-definite is a well-known result [14].

1.2. *k*-means++

k-means++ [15] is an algorithm originally envisioned as an initialisation for k-means clustering, designed to select a subset of highly dissimilar points from a corpus. Again, weights can easily be applied to each instance. The algorithm is as follows:

1) Choose an initial point at random from $\mathcal{X}$ with probabilities weighted by w. Remove the point from $\mathcal{X}$ and append to $\mathcal{A}$.

2) For each $x_i \in \mathcal{X}$, compute $D(x_i) = \min_{x' \in \mathcal{A}} \|x_i - x'\|$, the distance between x_i and the closest point in $\mathcal{A}$.

3) Choose the next point with probability $\propto w_i D(x_i)^2$.

4) Repeat steps 2 and 3 until k points are chosen.

1.3. Training strategy

Ideally, the samplers would have access to up-to-date feature embeddings for every draw. However, recomputing $\mathcal{X}$ (not to mention S) at every training iteration would be slow. Instead, we propose to only update the samplers periodically every t iterations; t is thus a trade-off between training speed and the quality of the similarity information in S. Where no pretrained feature extractor is available, the first t iterations are performed with standard (weighted) random samplers, although we posit that using the diversity-based samplers with features from a newly-initialised network with random weights would have an equivalent effect.

2. EXPERIMENTS

In this section, we evaluate the proposed method on a real-world domain shift problem, namely, the detection of humpback whale calls across data from different acoustic monitoring programs [8]. The dataset comprises 43,385 samples split roughly equally across 4 recording locations (Madagascar, UK, Hawaii, and Australia). Each sample is a PCEN-normalised [16] mel-spectrogram of a 4-second audio clip sampled at 10 kHz, labelled as either "humpback whale" or "not humpback whale". Some exemplar spectrograms are shown in Figure 1 (note, these images are linear-scaled and pre-PCEN). A simple 4-layer CNN architecture is used as the core model, with 16 filters per layer and RELU activations.

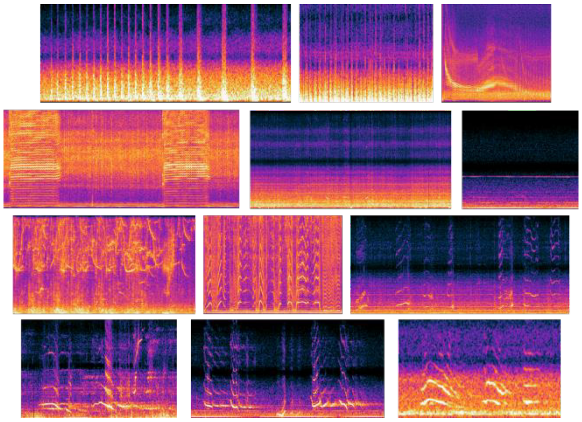


Figure 1: Some exemplar spectrograms of sounds in the dataset (5 kHz bandwidth, time axis scales variable). Top row: sperm whale clicks, pilot whale clicks, seal vocalisations. Second row: minke whale boings, right whale calls in strong vessel noise, electrical interference. Third row: dolphin whistles, dolphin creaks, right whale calls. Bottom row: three humpback whale calls.

Experiments are conducted using the DomainBed framework [1]. This means 3 locations ("domains") are used at a time for training and the remaining domain for testing. Models are trained for 2,000 iterations, with the samplers updated every 400. Hyperparameters are chosen via random search of size 40 using an oracle validation set (i.e., a set following the same distribution as the test set) as this provides the greatest stability for hyperparameter tuning and reduces the noise in the results from this source. Experiments are repeated 5 times for reproducibility, using different random seeds for hyperparameters, weight initialisations, and dataset splits. All other options follow the DomainBed defaults.

We use the DPPy Python package [17] for k-DPP sampling (specifically, the exact spectral sampler) and the scikit-learn [18] implementation of k-means++. The features used for sampling are the same features used by the distribution alignment methods (i.e., the activations from the last convolutional layer of the model). See [8] for more model and dataset details and [1] for further training and hyperparameter details.

Table 1: Test domain accuracy (%) for each sampler and training algorithm.

	DG			UDA		
Sampler	ERM	CORAL	DANN	CORAL	DANN	Average
Random	91.3 ± 0.7	86.2 ± 0.2	82.1 ± 1.2	90.4 ± 0.8	81.7 ± 1.6	86.3 ± 0.5
k-DPP	91.6 ± 0.4	90.6 ± 0.2	**87.7 ± 0.9**	**94.0 ± 0.2**	85.2 ± 1.1	89.8 ± 0.3
k-means++	**92.9 ± 0.7**	**91.5 ± 0.7**	87.3 ± 1.6	93.8 ± 0.2	**86.6 ± 1.7**	**90.4 ± 0.5**

2.1. Impact on generalisation performance

First, we compare the effect of the samplers on the generalisation power of the trained models. We do this for 2 invariance regularisation algorithms: correlation alignment (CORAL) [3] and domain-adversarial neural networks (DANN) [5], in both adaptive (UDA) and non-adaptive (domain generalisation, DG) paradigms. In the DG setting, these are used only to align the 3 training domains to each other. For UDA, in addition to this, the training domains are also aligned to an unlabelled, held-out subset of the test domain (that is, *not* the same samples that are used to determine accuracy, nor tune hyperparameters). We also test ERM, which does not explicitly perform domain alignment and by its nature is DG only.

In addition to our 2 proposed diversity-based data samplers (k-DPP and k-means++), we compare a baseline of standard class-weighted random sampling. Our performance metric is average model accuracy across the 4 test domains, reported in Table 1, along with the standard error across the 5 repeats.

Firstly, our results reproduce findings from previous work [1, 2, 8–10]: with standard random samplers, invariance regularisation performs poorly, underperforming ERM by as much as 10%. The results clearly show that using diversity-based sampling improves these methods, with consistent accuracy gains of 4 to 5 percentage points. Interestingly, ERM is also slightly improved, suggesting a general benefit to ensuring equal representation of all sound events.

Despite these gains, both CORAL and DANN still underperform ERM in the DG setting, showing just how difficult the DG problem is – as well as how strong the ERM baseline is. However, in the UDA setting, diversity-based sampling enables CORAL to finally exceed ERM, achieving the highest performance out of all the methods we test.

On average, accuracy is slightly higher with k-means++ than with the k-DPP, although this is within margin of uncertainty. In addition, k-means++ is computationally faster, easier to scale, and perhaps also more intuitive to understand and implement, making it the more favourable method overall.

So, we have shown that inducing diversity allows models trained with invariance regularisation to generalise better to new domains. In Section 1, we claimed that this is because diverse samples are more representative of their underlying distributions, and that this reduces error when estimating the distances between distributions. We test both parts of this claim next.

2.2. Improved distribution coverage

Recall our aim is to choose subsets $\mathcal{A}$ which are more representative of the full set $\mathcal{X}$. This can be recognised as the problem of vector quantisation. Thus, a measure of the "representativeness" of $\mathcal{A}$ is a low value of the quantisation error (QE)

$$\mathrm{QE} = \sum_{x_i \in \mathcal{X}} \min_{x' \in \mathcal{A}} \|x_i - x'\|^2 \,, \tag{3}$$

that is, the sum of squared distances between each x_i and the closest point in $\mathcal{A}$.

Table 2 compares the average QE of the 3 samplers over 1000 independent draws of $\mathcal{A}$ from each domain, with $k = 32$, and based on the features extracted by the ERM models from Section 2.1. The QE of the k-DPP and k-means++ are shown to be greatly reduced compared to the random sampler, by 36% and 65% respectively. Given the direct connection between (3) and the k-means++ selection criterion, it is perhaps unsurprising that the QE is so much lower for the latter, although this has not translated into greater generalisation power to the same extent.

Table 2: Average quantisation error for each sampler.

Sampler	QE
Random	6861 ± 23
k-DPP	4418 ± 12
k-means++	**2425 ± 6**

2.3. Lower-error distance estimates

Finally, we test the claim that diverse sampling improves distance estimation between distributions. To do this, we compare the estimation error of a popular distance estimate (the MMD) applied to the features of our multi-domain dataset.

Let $\mathcal{F} = \mathbb{R}^d$ be the feature space induced by our model. The MMD is computed on the basis of a positive-definite kernel $\kappa : \mathcal{F} \times \mathcal{F} \to \mathbb{R}$ and is defined as the distance between distribution means embedded in the reproducing kernel Hilbert space $\mathcal{H}$ associated with κ. For 2 distributions $\mathbb{P}_1, \mathbb{P}_2 \in \mathcal{P}$, we have

$$\mathrm{MMD}(\mathbb{P}_1, \mathbb{P}_2) = \|\mu(\mathbb{P}_1) - \mu(\mathbb{P}_2)\|_{\mathcal{H}}, \tag{4}$$

where $\mu : \mathcal{P} \to \mathcal{H}$ is the mean map operation

$$\mu(\mathbb{P}) = \mathbb{E}_{X\sim\mathbb{P}}[\phi(X)] \cong \frac{1}{n}\sum_{i=1}^{n} \phi(x_i) \tag{5}$$

and $\phi : \mathcal{F} \to \mathcal{H}$ is the implicit mapping associated with $\mathcal{H}$. It has been shown that for certain *characteristic* kernels, including the RBF, μ is injective, meaning every possible feature distribution $\mathbb{P} \in \mathcal{P}$ is uniquely represented in $\mathcal{H}$ and the MMD is 0 if and only if the distributions are identical [19].

Concretely, the method is as follows. We train a model by ERM on 3 domains at a time, as in Section 2.1. Our target is to compute the average pairwise MMD between these 3 domains, based on features extracted from the model and the same RBF mixture kernel defined in (2). We compute a set of 1000 MMDs using only 32 examples per domain, drawn stochastically using each of the 3 samplers. We measure the error of these w.r.t. a "ground-truth" MMD computed using all the available data (~8000 examples per domain). We can do this because (5) is a consistent

Table 3: Mean absolute percentage error (%) of the MMD estimates between domains, based on samples drawn by different sampling strategies.

	MAPE by held-out domain (%)				
Sampler	1	2	3	4	Average
Random	50.3 ± 2.1	**20.0 ± 0.5**	33.9 ± 1.8	28.4 ± 1.3	33.1 ± 0.8
k-DPP	28.5 ± 2.3	26.8 ± 2.2	**15.9 ± 0.8**	**20.4 ± 2.8**	**22.9 ± 1.1**
k-means++	**8.7 ± 0.8**	55.7 ± 1.1	26.5 ± 4.7	21.4 ± 4.2	28.1 ± 1.6

estimator of the embedded distribution mean: thus, an estimate computed with a larger sample will (in expectation) be closer to the true value of the MMD. Specifically, we compute the mean absolute percentage error (MAPE) in the MMDs, defined as

$$\mathrm{MAPE} = 100\% \frac{1}{1000D} \sum_{r=1}^{1000} \left| D - \widehat{D}_r \right| \tag{6}$$

where D is the "ground-truth" MMD computed using the full dataset and $\widehat{D}_r$ are the MMDs computed using only 32 examples per domain. As before, we do this for all 4 combinations of training domains, and repeat 5 times for reproducibility. The results are shown in Table 3.

The results show that both diversity-based samplers reduce the MAPE in the small-sample MMD estimates compared to the random sampler, for all but one of the training domain combinations. It is unclear to us why this pattern is reversed for Domain 2; however, the average over all domains is nonetheless favourable. In this case, we can see that the k-DPP has produced significantly better MMD estimates than k-means++ (despite having higher QE), but, again, this has not directly translated into higher model accuracy.

Overall, these results substantiate our hypothesis that the improved generalisation seen when using invariance regularisation is due to the higher-quality distance estimates generated by diverse samples, but this is of course by no means conclusive proof, and several peculiar phenomena in the results remain to be answered.

3. DISCUSSION

This paper introduced a novel use-case of diversity, in the form of enhancing the generalisation power of neural networks trained with invariance regularisation. We demonstrated that training on diverse minibatches enabled an adaptive invariance-regularised model to surpass the performance of ERM, a result that could not be achieved using standard random sampling methods. Our analysis supported the claim that this was due to the improved distance estimates attained by increasing the distribution coverage of the minibatches.

Regardless of the mechanism by which the performance gain occurs, the notion of a generalised balancing that is not bound by the available labels remains attractive, especially given that performance of the ERM-trained model also improved. It is interesting to note that inducing diversity tends to upweight the importance of outliers in the training set, which is at odds with a common notion in machine learning that outliers should in fact be removed. Specifying relevance or "quality" weights, as was done here for class weights, offers a way to regulate this trade-off. Further exploration of this in the contexts of domain generalisation, invariance regularisation, and their application to bioacoustic monitoring, would form a good basis for future work.

4. ACKNOWLEDGEMENTS

The authors acknowledge the use of the IRIDIS High Performance Computing Facility, and associated support services at the University of Southampton, in the completion of this work.

5. REFERENCES

[1] I. Gulrajani and D. Lopez-Paz, 'In Search of Lost Domain Generalization', *ICLR*, 2021.

[2] P. W. Koh *et al.*, 'WILDS: A Benchmark of in-the-Wild Distribution Shifts', *ICML*, 2021.

[3] B. Sun and K. Saenko, 'Deep CORAL: Correlation Alignment for Deep Domain Adaptation', *ECCV*, 2016.

[4] H. Li, S. J. Pan, S. Wang, and A. C. Kot, 'Domain Generalization with Adversarial Feature Learning', *CVPR*, 2018.

[5] Y. Ganin *et al.*, 'Domain-Adversarial Training of Neural Networks', *JMLR*, 2015.

[6] M. Long, Z. Cao, J. Wang, and M. I. Jordan, 'Conditional Adversarial Domain Adaptation', *Advances in Neural Information Processing Systems*, 2017.

[7] J. Shen, Y. Qu, W. Zhang, and Y. Yu, 'Wasserstein Distance Guided Representation Learning for Domain Adaptation', *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, 2017.

[8] A. Napoli and P. White, 'Unsupervised Domain Adaptation for the Cross-Dataset Detection of Humpback Whale Calls', in *DCASE*, 2023.

[9] A. Dubey, V. Ramanathan, A. Pentland, and D. Mahajan, 'Adaptive Methods for Real-World Domain Generalization', *CVPR*, 2021.

[10] I. Gao, S. Sagawa, P. W. Koh, T. Hashimoto, and P. Liang, 'Out-of-Distribution Robustness via Targeted Augmentations', *ICML*, 2023.

[11] A. Napoli and P. White, 'Diversity-Based Sampling for Imbalanced Domain Adaptation', *EUSIPCO*, 2024.

[12] A. Kulesza and B. Taskar, 'Determinantal point processes for machine learning', *Foundations and Trends in Machine Learning*, 2012.

[13] M. Outidrarine, P. Baudet, V. Lostanlen, M. Lagrange, and J. S. Ulloa, 'Exploring Eco-Acoustic Data with K-Determinantal Point Processes', *DCASE*, 2022.

[14] H. Wendland, *Scattered Data Approximation*. in Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2004.

[15] D. Arthur and S. Vassilvitskii, 'k-means++: The Advantages of Careful Seeding', *Proceedings of the*

eighteenth annual ACM-SIAM symposium on Discrete algorithms, 2007.

[16] Y. Wang, P. Getreuer, T. Hughes, R. F. Lyon, and R. A. Saurous, ‘Trainable Frontend For Robust and Far-Field Keyword Spotting’, *ICASSP*, 2016.

[17] G. Gautier, G. Polito, R. Bardenet, and M. Valko, ‘DPPy: DPP Sampling with Python’, *JMLR - Machine Learning Open Source Software*, 2019.

[18] F. Pedregosa *et al.*, ‘Scikit-learn: Machine Learning in Python’, *Journal of Machine Learning Research*, 2011.

[19] B. K. Sriperumbudur, K. Fukumizu, A. Gretton, G. R. G. Lanckriet, and B. Schölkopf, ‘Kernel Choice and Classifiability for RKHS Embeddings of Probability Distributions’, *Advances in Neural Information Processing Systems*, 2009.

TOYADMOS2#: YET ANOTHER DATASET FOR THE DCASE2024 CHALLENGE TASK 2 FIRST-SHOT ANOMALOUS SOUND DETECTION

Daisuke Niizumi, Noboru Harada, Yasunori Ohishi, Daiki Takeuchi, Masahiro Yasuda

NTT Corporation, Japan
daisuke.niizumi@ntt.com

ABSTRACT

First-shot anomalous sound detection (ASD) is a task designed to challenge a system's applicability to new data based on the needs of real-world application scenarios. This paper describes new ToyADMOS2 data to evaluate the first-shot compliant systems for the DCASE2024 Challenge Task 2, First-Shot Unsupervised Anomalous Sound Detection for Machine Condition Monitoring. The new data is designed to differ from the previous in the new machine sounds, including HoveringDrone, HairDryer, ToyCircuit, and ToothBrush, as well as in that each sound has a different background noise. The HairDryer and ToothBrush sounds are also designed as examples of ASD application scenarios in factory pre-shipment inspections, and we confirm their potential in the evaluation. We detail these data and show the baseline performance for reference in future studies.

Index Terms— DCASE 2024 Challenge Task 2, First-Shot Anomalous Sound Detection, ToyADMOS dataset

1. INTRODUCTION

Anomalous sound detection (ASD), which uses sound as a cue to detect anomalies, has been actively studied for applications such as factory automation. To facilitate the research, the annual Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge hosts an ASD task that has been drawing the attention of various participants.

The ASD challenges (the DCASE 2020-24 Challenge Task 2) [1, 2, 3, 4, 5] take a task setting that provides only normal samples for training while using normal and anomalous samples to evaluate the detection systems at test time (unsupervised ASD). This setting reflects the real-world situation where anomalous samples are hardly available or the available data cannot cover the distribution of the possible anomalies.

As we found new directions, the focus of the challenges transitioned from the ASD problem itself in 2020 [1] to a domain-shift condition in 2021 [2], a domain generalization in 2022 [3], and a first-shot condition in 2023 [4]. The first-shot condition reflects the demand for the rapid deployment of ASD systems for new machine sounds. While the conventional ASD setting uses the same machine sounds for both the development and evaluation phases, it uses only the new machine sounds for evaluation. In addition, it limits the use of training data other than the target machine sounds. Therefore, it forces the detection systems to do the first-shot solution on the new data.

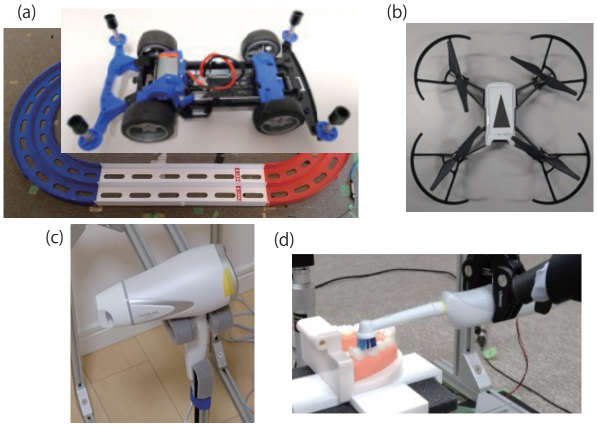


Figure 1: ToyADMOS2# includes two toys (a) ToyCircuit and (b) HoveringDrone and two home electrical appliances, (c) HairDryer, and (d) ToothBrush, bringing the evaluation setting closer to the real-world problem.

The 2024 challenge (DCASE2024T2) [5] extends the first-shot condition and limits the availability of the sample attribute information for some machines. This reflects the trend in ASD solutions where the outlier exposure (OE) approach [6, 7, 8] typically perform better. The OE approach uses sounds from different machines or attributes as anomalies; however, in real-world scenarios, we cannot always obtain machine attribute information.

For the 2020–2023 challenges, the developed datasets include: ToyADMOS [9], MIMII [10], ToyADMOS2 [11], MIMII DUE [12], MIMII DG [13], and ToyADMOS2+ [14]. This paper introduces new data for the ToyADMOS family, ToyADMOS2# (*two sharp*), that meets the first shot requirement of the DCASE2024T2.

The previous ASD sounds are mainly targeted at detecting failures in operating machines, such as production ma-

Table 1: ToyADMOS data history.

Machine type	DCASE usage	Product type	Mobility	Recorded sounds
(i) ToyADMOS/ToyADMOS2/ToyADMOS2+				
ToyCar	2020-24 dev. & 2020-22 eval.	Toy	Fixed	Running while fixed at a stand
ToyTrain	2020-24 dev. & 2020-22 eval.	Toy	Mobile	Circling on a railroad track
ToyConveyor	2020 dev. & eval.	Toy	Fixed	Conveying small cargo
Vacuum	2023 evaluation	Appliance	Fixed	Vacuuming at a fixed point
ToyTank	2023 evaluation	Toy	Mobile	Moving back and forth
ToyNscale	2023 evaluation	Toy	Mobile	Circling on a railroad track
ToyDrone	2023 evaluation	Toy	Mobile	Taking off and landing
(ii) ToyADMOS2#				
ToyCircuit	2024 evaluation	Toy	Mobile	Circling on a circuit track
HoveringDrone	2024 evaluation	Toy	Stationary	Hovering with rotation at a fixed point
HairDryer	2024 evaluation	Appliance	Fixed	Blowing at a fixed point
ToothBrush	2024 evaluation	Appliance	Fixed	Brushing at a fixed point

chines in a factory. In contrast, to show the possibility of an ASD application to the production process in a factory, we have two sound settings of typical home electrical appliances as examples of pre-shipment inspection. ToyADMOS2# provides the additional training and evaluation datasets of the DCASE2024T2 with four machine sounds described in Fig. 1. The dataset is available at Zenodo [5, 15, 16].

2. PREVIOUS TOYADMOS DATASETS

The ToyADMOS family has released three datasets, and Table 1(i) lists all the data they provided to the past DCASE Challenge Task 2. The first release, ToyADMOS [9], contains three miniature machine (toy) sounds with various anomalous sounds. It uses toys as machine sound sources and simulates anomalies by breaking parts of them to address the difficulty of collecting anomalous sounds. ToyADMOS2 [11], released in 2021, contains a wider variety of sounds of two toys and enables the generation of datasets that simulate domain shift conditions. ToyADMOS2+ [14] recorded the sounds of new machines to enable first-shot ASD and provided them as an evaluation set in the DCASE 2023 Challenge Task 2 [17]. It also features a home electrical appliance as one of the machines, introducing a new ASD setting of the everyday sounds around us.

3. TOYADMOS2#: NEW DATA FOR THE DCASE2024 CHALLENGE TASK 2

ToyADMOS2# adds data for the four machines shown in Table 1(ii) for evaluation under the new first-shot ASD condition. To distinguish them from the previous data, we used the sounds of two home electrical appliances and different background noises for all of them. We specifically designed HairDryer and ToothBrush as example scenarios of ASD applications in product pre-shipment inspections of home electrical appliances.

ToyCircuit: The sound of the ToyCar (TAMIYA Mini 4WD) driving on a circuit track, characterized by the sound of friction with the track lanes and the change in distance from the microphone.

HoveringDrone: The sound of the drone (DJI Tello) hovering at one point and rotating. Unlike the ToyDrone last year, we made the distance from the microphone almost constant.

HairDryer: The sound of the dryer (Panasonic/Koizumi) airflow to evaluate the detection of anomalous sound when airflow is obstructed, such as when unintended foreign matter adheres to the dryer during the production process.

ToothBrush: The sound of the electric toothbrush (Brown DB5510) brushing teeth to evaluate the detection of anomalies, such as brushes manufactured with defects. We maintained constant pressure between the brush against the teeth to avoid changes in sound due to pressure differences.

Table 2 summarizes the details of each machine, especially the speed/mode and background noise characterizing the differences between them. Domain shift settings commonly changed from source to target: ID from A and B to C, and microphone from 1 to 2. We changed the speed and mode for each machine and basically assigned the unused values in the source to the target.

Table 3 details the anomalous conditions for each machine. ToyCircuit differs from ToyCar in that the anomalous sounds are also produced by friction with the running surface. HoveringDrone assumes that the adhesion of foreign objects occurs during use. Anomalous conditions for HairDryer and ToothBrush also assume adhesion while further assuming that defects in the manufacturing process of these products cause anomalous sounds and that future ASD systems detect them in product pre-shipment inspections. For example, future ASD systems could be combined with optical inspection (e.g., toothbrushes in the MVTec AD dataset [18]) to improve factory product inspection performance. Fig. 2 showcases the anomalous condition examples of ToothBrush used in the recordings.

3.1. Recording control details

The sounds were recorded in a controlled environment by following the recording layouts and microphone arrange-

Table 2: ToyADMOS2# data details.

	Settings and parameter variations				Domain shift settings	Samples	
Machine type	ID	Dur.	Speed/mode	Background noise	Source→target	Training	Eval.
(a) ToyCircuit	A, B, C	8 s	1: 1.3 V, 2: 1.4 V, 3: 1.5 V, 4: 1.6 V	Large air conditioner outdoor unit outlet noise	ID: A,B→C, Mic: 1→2 Speed: 1, 2, 3→1, 4	Src. 990 Trg. 10	Src. 100 Trg. 100
(b) HoveringDrone[†]	A, B, C	8 s	1: Rotate CW 180° and CCW 180°, 2: Rotate CW 180° and CW 180°, 3: Rotate CCW 180° and CCW 180°	City noise near a river under a highway bridge	ID: A,B→C, Mic: 1→2 Mode: 1, 2→3	Src. 990 Trg. 10	Src. 100 Trg. 100
(c) HairDryer	A, B, C	7 s	1: 92 V, 2: 96 V, 3: 100 V, 4: 104 V	Running water sound in a drainage ditch in a park	ID: A,B→C, Mic: 1→2 Speed: 2,3→1,4	Src. 990 Trg. 10	Src. 100 Trg. 100
(d) ToothBrush[†]	A, B, C	6 s	1: Lower teeth/ 2.7 V, 2: Lower teeth/ 2.8 V, 3: Upper teeth/ 2.8 V, 4: Lower teeth/ 2.9 V, 5: Upper teeth/ 2.9 V, 6: Lower teeth/ 3.0 V	Home air purifier outlet noise	ID: A,B→C, Mic: 1→2 Mode: 2, 3, 4, 5→1, 6	Src. 990 Trg. 10	Src. 100 Trg. 100

[†] The actual parameters (sample attributes) were not provided in the data files following the focus of the DCASE2024T2.

Table 3: Anomaly conditions for each machine type.

(a) ToyCircuit		(b) HoveringDrone	
Part	Condition	Part	Condition
Tire	Foreign objects	Arm	Foreign object
	Scratches	Propeller	Foreign object/one side
Shaft	No grease		Foreign object/two sides
Gear	Locked gear	Body	Offset weight

(c) HairDryer		(d) ToothBrush	
Part	Condition	Part	Condition
Outlet	Foreign object	Brush hair	Damaged brush hair
Inlet	Foreign object		Foreign object stuck
Vane	Foreign object		Partially missing brush hair
	Chipped vane	Brush head	Half-insertion of brush head

ments shown in Figs. 3 and 4 and by automating the machines' controls. The system used optical sensors to manage the laps of the ToyCircuit, image recognition to control the HoveringDrone, and automatic control of the main power supply for the appliances. The resulting sounds should reflect the differences in hardware, actual mechanical movements, and course and drive/flight conditions.

While in the controlled sound recording environment, we limited the number of samples obtained in a single recording to make the data distribution closer to the intrinsic nature of the machine's data distribution. In particular, the recording of the AC-powered HairDryer can continuously provide many samples at once; however, it ends up with many similar samples and cannot cover the data distribution gained by the differences in installation, assembly, time, and natural degradation. Therefore, we avoided continuous operation and switched to recording under physically different conditions for no more than 30 samples.

3.2. Data sample details

All the operating sound and noise samples were recorded with 48-kHz sampling, 24 bits for each channel, and then downsampled to 16-kHz, 16 bits, monaural in the final data samples. Sample duration varied from 6 s to 8 s, depending on the machine type, as shown in Table 2.

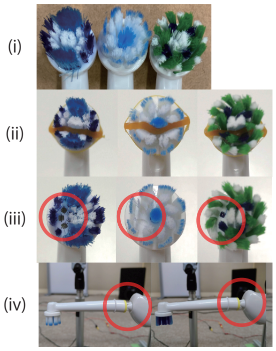


Figure 2: Anomalous condition examples of ToothBrush: (i) Damaged brush hair, (ii) foreign object stuck in the brush, (iii) partially missing brush hair, and (iv) half-insertion of the brush head.

The training data (Additional training dataset) for each machine type has 1000 normal samples, 990 from the source domain and 10 from the target domain. The evaluation data (Evaluation dataset) for each machine type consists of 50 normal and 50 anomaly samples from each source and target domain, for a total of 200 samples. The total of these data provides 4800 samples with 580 minutes of recordings. The data are available at the Zenodo links [15, 16] under the Creative Commons Attribution 4.0 International Public License [19].

4. BENCHMARKS

We show the evaluation results obtained using the DCASE2024 Challenge Task 2 baseline system in Table 4. The baseline is a reconstruction-based ASD system using Autoencoder and has two operating modes: First-shot-compliant Simple Autoencoder mode and Selective Mahalanobis Autoencoder mode. The former calculates the distance between the input sample and the reconstruction using MSE (mean squared error), while the latter does based on Mahalanobis's distance [20]. The results are the area under the receiver operating characteristic curve (AUC) and partial AUC (pAUC), where the pAUC measures performance in a

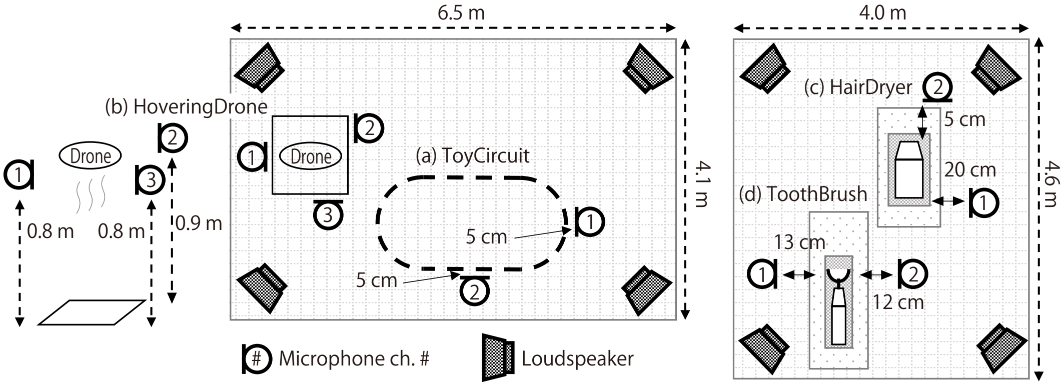


Figure 3: Recording-room layouts and microphone arrangements: (a) ToyCircuit, (b) HoveringDrone, (c) HairDryer, and (d) ToothBrush.

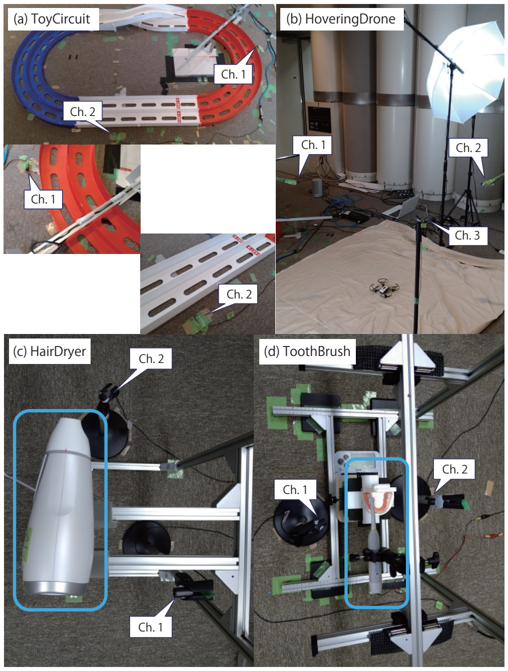


Figure 4: Microphone arrangements: (a) ToyCircuit, (b) HoveringDrone, (c) HairDryer, and (d) ToothBrush.

low false-positive rate (FPR) range $[0, p]$ with a p of 0.1. For the details, see [20, 21].

The results show that for all machine types, the baseline performs well in the source domain while generalization to the target domain is difficult, a trend similar to that for data through 2023. The exception for ToothBrush is that the baseline also performs well on the target domain data, suggesting that the degree of domain shift is small. The performance of the machines simulating a product pre-shipment inspection scenario (HairDryer and ToothBrush) shows a similar trend to that of the other two machines, implying the potential for future ASD applications in the scenario.

Table 4: Benchmark results

Machine type	AUC [%]		pAUC [%]
	Source	Target	
(i) First-shot-compliant baseline: Simple Autoencoder mode			
ToyCircuit	77.50 ± 0.82	51.25 ± 1.22	50.12 ± 0.17
HoveringDrone	85.93 ± 0.77	47.87 ± 4.30	51.05 ± 1.46
HairDryer	64.94 ± 3.40	43.75 ± 2.09	50.56 ± 1.01
ToothBrush	73.80 ± 1.08	70.14 ± 4.92	54.19 ± 1.73
(ii) First-shot-compliant baseline: Selective Mahalanobis Autoencoder mode			
ToyCircuit	69.67 ± 1.72	42.34 ± 2.11	49.23 ± 0.03
HoveringDrone	84.07 ± 1.10	48.50 ± 3.64	58.95 ± 2.76
HairDryer	64.23 ± 3.44	56.71 ± 1.97	55.12 ± 0.71
ToothBrush	63.17 ± 2.43	57.55 ± 3.59	49.81 ± 1.29

5. CONCLUSION

This paper introduced new ToyADMOS2 data to evaluate the first-shot compliant systems for the DCASE2024 Challenge Task 2, First-Shot Unsupervised Anomalous Sound Detection for Machine Condition Monitoring. The first-shot anomalous sound detection (ASD) is a task designed to challenge a system's applicability to new data based on the needs of real-world application scenarios. The new sounds include HoveringDrone, HairDryer, ToyCircuit, and ToothBrush, and they are mixed with four different environmental noises to enhance their differences from the previous sounds. We specifically designed two sounds, HairDryer and ToothBrush, as example scenarios of ASD applications in product pre-shipment inspections of home electrical appliances and confirmed their potential in the evaluation. The ToyADMOS2# dataset (DCASE 2024 Challenge Task 2 Additional Training Dataset and Evaluation Dataset) is available at [5, 15, 16] with the Creative Commons Attribution 4.0 International Public License [19].

6. ACKNOWLEDGMENT

The authors thank Hideaki Sanada, Akito Karasawa, Keisuke Hasegawa, Azusa Takahashi, and Yuri Musashijima for their hard work implementing the systems and conducting experiments.

7. REFERENCES

[1] Y. Koizumi, Y. Kawaguchi, K. Imoto, T. Nakamura, Y. Nikaido, R. Tanabe, H. Purohit, K. Suefusa, T. Endo, M. Yasuda, and N. Harada, "Description and discussion on DCASE2020 challenge task2: Unsupervised anomalous sound detection for machine condition monitoring," in *DCASE Workshop*, November 2020, pp. 81–85.

[2] Y. Kawaguchi, K. Imoto, Y. Koizumi, N. Harada, D. Niizumi, K. Dohi, R. Tanabe, H. Purohit, and T. Endo, "Description and discussion on dcase 2021 challenge task 2: Unsupervised anomalous detection for machine condition monitoring under domain shifted conditions," in *DCASE Workshop*, Barcelona, Spain, November 2021, pp. 186–190.

[3] K. Dohi, K. Imoto, N. Harada, D. Niizumi, Y. Koizumi, T. Nishida, H. Purohit, R. Tanabe, T. Endo, M. Yamamoto, and Y. Kawaguchi, "Description and discussion on dcase 2022 challenge task 2: Unsupervised anomalous sound detection for machine condition monitoring applying domain generalization techniques," in *DCASE Workshop*, Nancy, France, November 2022.

[4] K. Dohi, K. Imoto, N. Harada, D. Niizumi, Y. Koizumi, T. Nishida, H. Purohit, R. Tanabe, T. Endo, and Y. Kawaguchi, "Description and discussion on DCASE 2023 challenge task 2: First-shot unsupervised anomalous sound detection for machine condition monitoring," in *DCASE Workshop*, Tampere, Finland, September 2023, pp. 31–35.

[5] "DCASE 2024 challenge task 2: First-shot unsupervised anomalous sound detection for machine condition monitoring," 2024, https://dcase.community/challenge2024/task-first-shot-unsupervised-anomalous-sound-detection-for-machine-condition-monitoring.

[6] D. Hendrycks, M. Mazeika, and T. Dietterich, "Deep anomaly detection with outlier exposure," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=HyxCxhRcY7

[7] R. Giri, S. V. Tenneti, F. Cheng, K. Helwani, U. Isik, and A. Krishnaswamy, "Self-supervised classification for detecting anomalous sounds," in *DCASE Workshop*, 2020, pp. 46–50.

[8] K. Wilkinghoff and F. Kurth, "Why do angular margin losses work well for semi-supervised anomalous sound detection?" *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 608–622, 2024.

[9] Y. Koizumi, S. Saito, H. Uematsu, N. Harada, and K. Imoto, "ToyADMOS: A dataset of miniature-machine operating sounds for anomalous sound detection," in *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, November 2019, pp. 308–312.

[10] H. Purohit, R. Tanabe, T. Ichige, T. Endo, Y. Nikaido, K. Suefusa, and Y. Kawaguchi, "MIMII Dataset: Sound dataset for malfunctioning industrial machine investigation and inspection," in *DCASE Workshop*, November 2019, pp. 209–213.

[11] N. Harada, D. Niizumi, D. Takeuchi, Y. Ohishi, M. Yasuda, and S. Saito, "ToyADMOS2: Another dataset of miniature-machine operating sounds for anomalous sound detection under domain shift conditions," in *DCASE Workshop*, Barcelona, Spain, November 2021, pp. 1–5.

[12] R. Tanabe, H. Purohit, K. Dohi, T. Endo, Y. Nikaido, T. Nakamura, and Y. Kawaguchi, "MIMII DUE: Sound dataset for malfunctioning industrial machine investigation and inspection with domain shifts due to changes in operational and environmental conditions," *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 21–25, 2021.

[13] K. Dohi, T. Nishida, H. Purohit, R. Tanabe, T. Endo, M. Yamamoto, Y. Nikaido, and Y. Kawaguchi, "MIMII DG: Sound dataset for malfunctioning industrial machine investigation and inspection for domain generalization task," in *DCASE Workshop*, Nancy, France, November 2022.

[14] N. Harada, D. Niizumi, D. Takeuchi, Y. Ohishi, and M. Yasuda, "ToyADMOS2+: New toyadmos data and benchmark results of the first-shot anomalous sound event detection baseline," in *DCASE Workshop*, Tampere, Finland, September 2023, pp. 41–45.

[15] "DCASE 2024 Challenge Task 2 Additional Training Dataset," 2024, https://zenodo.org/records/11259435.

[16] "DCASE 2024 Challenge Task 2 Evaluation Dataset," 2024, https://zenodo.org/records/11363076.

[17] "DCASE 2023 challenge task 2: First-shot unsupervised anomalous sound detection for machine condition monitoring," 2023, https://dcase.community/challenge2023/task-first-shot-unsupervised-anomalous-sound-detection-for-machine-condition-monitoring.

[18] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "MVTec AD — a comprehensive real-world dataset for unsupervised anomaly detection," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9584–9592.

[19] "Creative commons attribution 4.0 international public license," https://creativecommons.org/licenses/by/4.0/legalcode.

[20] N. Harada, D. Niizumi, Y. Ohishi, D. Takeuchi, and M. Yasuda, "First-shot anomaly sound detection for machine condition monitoring: A domain generalization baseline," in *2023 31st European Signal Processing Conference (EUSIPCO)*, 2023, pp. 191–195.

[21] N. Harada, Y. Musashijima, and D. Niizumi, "dcase2023_task2_baseline_ae," 2023, https://github.com/nttcslab/dcase2023_task2_baseline_ae.

DESCRIPTION AND DISCUSSION ON DCASE 2024 CHALLENGE TASK 2: FIRST-SHOT UNSUPERVISED ANOMALOUS SOUND DETECTION FOR MACHINE CONDITION MONITORING

Tomoya Nishida[1], *Noboru Harada*[2], *Daisuke Niizumi*[2], *Davide Albertini*[3], *Roberto Sannino*[3], *Simone Pradolini*[3], *Filippo Augusti*[3], *Keisuke Imoto*[4], *Kota Dohi*[1], *Harsh Purohit*[1], *Takashi Endo*[1], *and Yohei Kawaguchi*[1]

[1] Hitachi, Ltd., Japan, tomoya.nishida.ax@hitachi.com
[2] NTT Corporation, Japan, noboru.harada.pv@hco.ntt.co.jp
[3] STMicroelectronics, Switzerland,
[4] Doshisha University, Japan, keisuke.imoto@ieee.org

ABSTRACT

We present the task description of the Detection and Classification of Acoustic Scenes and Events (DCASE) 2024 Challenge Task 2: “First-shot unsupervised anomalous sound detection (ASD) for machine condition monitoring”. Continuing from last year’s DCASE 2023 Challenge Task 2, we organize the task as a first-shot problem under domain generalization required settings. The main goal of the first-shot problem is to enable rapid deployment of ASD systems for new kinds of machines without the need for machine-specific hyperparameter tunings. For the DCASE 2024 Challenge Task 2, sounds of new machine types were collected and provided as the evaluation dataset. In addition, attribute information such as the machine operation conditions were concealed for several machine types to simulate situations where such information are unavailable. We received 96 submissions from 27 teams, and an analysis of these submissions has been made in this paper. Several novel approaches, such as new ways of utilizing pre-trained models and pseudo-label classification approaches, have been used to beat the baseline system.

Index Terms— anomaly detection, acoustic condition monitoring, domain shift, first-shot problem, DCASE Challenge

1. INTRODUCTION

Anomalous sound detection (ASD) [1–7] is the task of identifying whether the sound emitted from a target machine is normal or anomalous. This leads to automatic detection of mechanical failures, which is vital in the fourth industrial revolution with AI-based factory automation. Using machine sounds for prompt detection of machine anomalies is useful for machine condition monitoring.

A major challenge concerning the application of ASD systems is that both the number and variety of anomalous samples can be inadequate in training. In 2020, we held the first ASD task in Detection and Classification of Acoustic Scenes and Event (DCASE) Challenge 2020 Task 2 [8]; “unsupervised ASD” which aimed to detect unknown anomalous sounds using only normal sound samples as training data. Following this task, handling of domain shifts was additionally tackled in the DCASE Challenge 2021 Task 2 [9] and 2022 Task 2 [10] for the wide spread application of ASD systems. Domain shifts are differences between the data in the source and target domains, which are caused by shifts in the operational conditions of the machine or environmental noise. The DCASE Challenge 2021 Task 2 [9] mainly focused on the use of domain adaptation techniques, whereas the DCASE Challenge 2022 Task 2 [10] focused on the use of domain generalization techniques.

In the DCASE Challenge 2023 Task 2 [11], ”first-shot unsupervised ASD,” real-world scenarios were explored even further as a ”first-shot” ASD task. This is a task that requires solving UASD against completely novel machine types, without access to data from similar machine types that can be used for model training or hyperparameter tuning. This scenario is typically encountered in real-world situations where the rapid deployment of ASD systems is required and collecting a variety of training or test data is infeasible. To realize this problem setting, the evaluation dataset was created by completely new machine types unseen in the development dataset. This setup prevented participants from performing handcrafted tunings which are difficult to implement in many real-world applications. For example, hyperparameter tuning for each machine type using the development dataset or training ASD systems with the same machine type sounds has become infeasible.

To further deepen the techniques that are useful for this problem setting grounded on real-world scenarios, we designed the DCASE Challenge 2024 Task 2 ”First-shot unsupervised anomalous sound detection for machine condition monitoring” by closely aligning to the problem setting established in the previous year. The main modifications from DCASE 2023 Task 2 are that the evaluation dataset is updated with new machine types unseen in the previous DCASE ASD challenges, and that attribute information such as the machine operation conditions are concealed for several machine types. The second modification concerns situations where such information is unavailable, with the aim of expanding the range of applicable scenarios in real-world settings.

2. FIRST-SHOT UNSUPERVISED ANOMALOUS SOUND DETECTION UNDER DOMAIN SHIFTED CONDITIONS

Let the L-sample time-domain observation $\boldsymbol{x} \in \mathbb{R}^L$ be an audio clip that includes sounds emitted from a machine. The goal of the ASD task is to determine a given machine as either normal or anomalous by computing an anomaly score $\mathcal{A}_\theta(\boldsymbol{x})$ using an anomaly score calculator $\mathcal{A}$ with parameters θ. The input of $\mathcal{A}$ can be the audio clip $\boldsymbol{x}$ or $\boldsymbol{x}$ with additional information such as labels indicating the operation condition of the machine. The machine is then determined to be anomalous when $\mathcal{A}_\theta(\boldsymbol{x})$ exceeds a pre-defined threshold ϕ as

$$\text{Decision} = \begin{cases} \text{Anomaly} & (\mathcal{A}_\theta(\boldsymbol{x}) > \phi) \\ \text{Normal} & (\text{otherwise}). \end{cases} \tag{1}$$

The primary difficulty in this task is to train the anomaly score calculator with only normal sounds (unsupervised ASD). The DCASE 2020 Challenge Task 2 [8] was designed to address this issue, and all the following tasks stand on this unsupervised ASD setting.

The domain-shift problem also needs to be solved for practical applications of ASD. Domain shifts are variations in conditions between training and testing phases that change the distribution of the observed sound data. These shifts can arise from differences in operating speed, machine load, heating temperature, environmental noise, microphone arrangement, and other factors. Two domains, the **source domain** and the **target domain**, are defined: the former refers to the original condition with sufficient training data and the latter refers to another condition with only a few samples. This year's task follows the 2022 and 2023 Task 2 [10, 11] setting, where the domain information is assumed to be unknown in the test phase and anomalies from both domains have to be detected with a single threshold. In this case, domain generalization is required to achieve good performance.

To further pursue the rapid development of ASD systems in real-world scenarios, solving ASD (a) against completely novel machine types (b) with only one section of training data (c) without handcrafted tunings that depend on test data, are highly important. This is because in real-world scenarios, customers may only possess a single novel machine, and collecting test data for handcrafted tuning may be infeasible. This problem setting was named as the "first-shot problem", and the 2023 Task 2 [11] was organized based on this problem setting. Specifically, the first-shot problem was realized by adding two features to the dataset: (i) Completely different sets of machine types between the development and evaluation dataset and (ii) Only one section for each machine type. Note that until 2022 Task 2, the data provided included multiple sections for each machine type, with the development and evaluation datasets sharing the same machine types.

While solving the first-shot problem under the domain generalization setting should be sufficient for many real-world applications, the results from the previous year suggested that there is still potential for further improvement in the solutions [11]. For this reason, we designed the DCASE Challenge 2024 Task 2, "First-shot unsupervised anomalous sound detection for machine condition monitoring" by closely aligning to the problem setting designed in the previous year. The main modifications from 2023 Task 2 are that the evaluation dataset consists of newly recorded sounds of new machine types and that attribute information are concealed for several machine types. By mostly following the same problem setting as in DCASE 2023 Task 2, the organizers aim to further deepen the techniques that are useful for first-shot ASD.

3. TASK SETUP

3.1. Dataset

The data for this task comprises three datasets: **development dataset**, **additional training dataset**, and **evaluation dataset**. The development dataset includes seven machine types, whereas the additional and evaluation dataset includes nine machine types, each having one section per machine type. **Machine type** means the type of machine such as fan, gearbox, etc. **Section** is a subset or whole data within each machine type.

Each recording is a single-channel audio with a duration of 6 to 10 s and a sampling rate of 16 kHz. We mixed machine sounds recorded at laboratories with environmental noise recorded at factories and in the suburbs to create each sample in the dataset. For the details of the recording procedure, please refer to the papers on ToyADMOS2 [12] and MIMII DG [13].

The **development dataset** consists of seven machine types (fan, gearbox, bearing, slide rail, valve, ToyCar, ToyTrain), and each machine type has one section that contains a complete set of the training and test data. Each section provides (i) 990 normal clips from a source domain for training, (ii) 10 normal clips from a target domain for training, and (iii) 100 normal clips and 100 anomalous clips from both domains for the test. We provided domain information (source/target) in the test data for the convenience of participants. For four machine types (fan, bearing, valve, ToyCar), attributes that represent operational or environmental conditions are also provided in the file names and attribute csvs. For the other three machine types, attributes are concealed. The **additional training dataset** provides novel nine machine types (3D-printer, air compressor, brushless motor, hairdryer, hovering drone, robotic arm, scanner, toothbrush, ToyCircuit). Each section consists of (i) 990 normal clips in a source domain for training and (ii) 10 normal clips in a target domain for training. For five machine types (3D-printer, hairdryer, robotic arm, scanner, ToyCircuit), attributes are provided in this dataset. For the other four machine types, attributes are concealed. The **evaluation dataset** provides the test clips that correspond to the additional training dataset, e.g. data of the same machine types as the additional training dataset. Each section consists of 200 test clips, none of which have a condition label (i.e., normal or anomaly), domain information, or attribute information.

Participants must train a model for a new machine type using only one section per machine type, without hyperparameter tuning using test datasets obtained from the same machine type, and for some of the machine types, without utilizing attribute information.

3.2. Evaluation metrics

We used the area under the receiver operating characteristic curve (AUC) to evaluate overall detection performance and the partial AUC (pAUC) to measure performance in a low false-positive rate range $[0, p]$, where we set $p = 0.1$. To evaluate each system under the domain generalization setting, we compute the AUC for each domain and pAUC for each section as

$$\text{AUC}_{m,n,d} = \frac{1}{N_d^- N_n^+} \sum_{i=1}^{N_d^-} \sum_{j=1}^{N_n^+} \mathcal{H}(\mathcal{A}_\theta(x_j^+) - \mathcal{A}_\theta(x_i^-)), \tag{2}$$

$$\text{pAUC}_{m,n} = \frac{1}{\lfloor pN_n^- \rfloor N_n^+} \sum_{i=1}^{\lfloor pN_n^- \rfloor N_n^+} \sum_{j=1}^{N_n^+} \mathcal{H}(\mathcal{A}_\theta(x_j^+) - \mathcal{A}_\theta(x_i^-)), \tag{3}$$

where m and n represent the index of a machine type and a section respectively, $d \in \{\text{source}, \text{target}\}$ represents a domain, $\lfloor \cdot \rfloor$ is the flooring function, and $\mathcal{H}(y)$ returns 1 when $y > 0$ and 0 otherwise. Here, $\{x_i^-\}_{i=1}^{N_d^-}$ are the normal test clips in domain d in section n of machine type m and $\{x_j^+\}_{j=1}^{N_n^+}$ are all the anomalous test clips in section n of machine type m. N_d^-, N_n^-, N_n^+ represent the number of normal test clips in domain d, normal test clips in section n, and anomalous test clips in section n, respectively.

The official score Ω is given by the harmonic mean of the AUC and pAUC scores overall machine types and sections:

$$\begin{aligned}\Omega \;=\;& h\left\{\mathrm{AUC}_{m,n,d},\ \mathrm{pAUC}_{m,n} \;\mid\right. \\ & \left. m \in \mathcal{M},\ n \in \mathcal{S}(m),\ d \in \{\text{source}, \text{target}\}\right\}, \quad (4)\end{aligned}$$

where $h\{\cdot\}$ represents the harmonic mean, $\mathcal{M}$ is the set of given machine types, and $\mathcal{S}(m)$ represents the set of sections for machine type m. Specifically, $\mathcal{S}(m) = \{00\}$ for the dataset in 2024.

3.3. Baseline systems and results

The task organizers provide a baseline system based on Autoencoders (AEs), featuring two distinct operating modes. This baseline system is the same system employed as the baseline in 2023 Task 2. Although both modes employ Autoencoder for training, they diverge in the computation of anomaly scores. In this paper, we introduce the baseline system along with its detection performance. For further information, please refer to [14].

3.3.1. Autoencoder training

The AE is first trained for both operating modes. First, the log-mel-spectrograms of each training sound clips $X = [X_1, \ldots, X_T]$ are calculated, where $X_t \in \mathbb{R}^F$ for $t = 1, \ldots, T$ are the frame-wise feature vectors at frame t, $F = 128$ is the number of mel-filters and T is the number of time-frames. For the input of the AE, $P = 5$ consecutive frames taken from X are concatenated as $\psi_t = [X_t^\mathsf{T}, \ldots, X_{t+P-1}^\mathsf{T}]^\mathsf{T} \in \mathbb{R}^D$ for each t, where $D = P \times F = 640$. The model parameters are optimized by minimizing the mean squared error (MSE) between the input ψ_t and the reconstructed output $r_\theta(\psi_t)$ for all inputs created from the training data.

3.3.2. Simple Autoencoder mode

In this mode, the anomaly score is calculated as the average of the MSE for all input features created from that sound clip, e.g.,

$$A_\theta(X) = \frac{1}{DK}\sum_{k=1}^{K}\|\psi_k - r_\theta(\psi_k)\|_2^2, \tag{5}$$

where $K = T - P + 1$, and $\|\cdot\|_2$ represents ℓ_2 norm.

3.3.3. Selective Mahalanobis mode

In this mode, the Mahalanobis distance between the system input and reconstructed feature is used to calculate the anomaly score. The anomaly score is given as

$$A_\theta(X) = \frac{1}{DK}\sum_{k=1}^{K}\min\{D_s(\psi_k, r_\theta(\psi_k)), D_t(\psi_k, r_\theta(\psi_k))\}, \tag{6}$$

$$D_s(\cdot) = \text{Mahalanobis}(\psi_k, r_\theta(\psi_k), \Sigma_s^{-1}), \tag{7}$$

$$D_t(\cdot) = \text{Mahalanobis}(\psi_k, r_\theta(\psi_k), \Sigma_t^{-1}), \tag{8}$$

where Σ_s^{-1} and Σ_t^{-1} are the covariance matrices of $r_\theta(\psi_k) - \psi_k$ for the source and target domain data of each machine type, respectively.

3.3.4. Results

Tables 1 show the AUC and pAUC scores for the two baselines on the development dataset. The average and standard deviations of the scores from five independent trials of training and testing are shown in the tables.

Table 1: Baseline results for development dataset.

Machine type	Mode	AUC [%]		pAUC [%]
		Source	Target	
ToyCar	MSE	66.98 ± 0.89	33.75 ± 0.81	48.77 ± 0.13
	MAHALA	63.01 ± 2.12	37.35 ± 0.83	51.04 ± 0.16
ToyTrain	MSE	76.63 ± 0.22	46.92 ± 0.80	47.95 ± 0.09
	MAHALA	61.99 ± 1.79	39.99 ± 1.37	48.21 ± 0.05
bearing	MSE	62.01 ± 0.64	61.40 ± 0.26	57.58 ± 0.32
	MAHALA	54.43 ± 0.27	51.58 ± 1.73	58.82 ± 0.13
fan	MSE	67.71 ± 0.70	55.24 ± 0.91	57.53 ± 0.19
	MAHALA	79.37 ± 0.44	42.70 ± 0.26	53.44 ± 1.03
gearbox	MSE	70.40 ± 0.58	69.34 ± 0.82	55.65 ± 0.44
	MAHALA	81.82 ± 0.33	74.35 ± 1.21	55.74 ± 0.35
slider	MSE	66.51 ± 1.66	56.01 ± 0.29	51.77 ± 0.35
	MAHALA	75.35 ± 3.02	68.11 ± 0.63	49.05 ± 1.00
valve	MSE	51.07 ± 0.88	46.25 ± 1.30	52.42 ± 0.50
	MAHALA	55.69 ± 1.44	53.61 ± 0.19	51.26 ± 0.47

4. CHALLENGE RESULTS

We received 96 submissions from 27 teams. Ten teams outperformed the simple Autoencoder baseline, and eleven outperformed the selective Mahalanobis baseline, which indicates the difficulty of the task. The number of teams outperforming the baselines was also close to that in 2023's task. Figure 1 shows the AUC values for the top 10 teams. In the source domain, many teams successfully improved the AUC values for half of the machine types, but showed lower AUC values than the baseline in the other half. As a result, the harmonic mean of the AUC values in the source domain was very close to the baselines for all teams. In the target domain, most of the top ten teams outperformed the baselines in most machine types. The order of the harmonic mean in the target domain was mostly aligned with the order of the official ranks, which means the performance on this domain was the key to achieve higher ranks.

Figure 2 compares the AUC values of the top 20 teams between the development and evaluation datasets. As can be seen, achieving high AUC values in the development dataset does not necessarily imply high AUC values in the evaluation dataset. This trend is seen especially in the first shot problem setting; The correlation coefficients between the mean AUCs of the development and evaluation dataset were higher for non-first shot tasks, e.g., 0.82 for 2021 and 0.83 for 2022, and lower for the first shot tasks, e.g., 0.62 for 2023 and 0.14 for 2024. This clarifies the difficulty of the first shot problem setting. As a result, teams that achieved high AUC values in the evaluation dataset (especially in the target domain, as noted above,) achieved higher ranks. Finally, in Figure 3, we compare the AUC values of the top 20 teams between machine types in which attribute information was provided and those in which attribute information was concealed. The number of teams that beat the baseline only for attribute-concealed machines (1) was fewer than that for attribute-available machines (6), which reveals that hiding the attribute has made the problem more challenging to some extent. Nevertheless, many high-ranking teams were able to surpass the baselines for both groups, indicating that those teams' solutions were capable of handling this new problem setting.

We summarize approaches used by top-ranked teams below.

a. Use of appropriate pre-trained models with fine-tunings

Using classification tasks such as machine type, domain or attribute classification as an auxiliary task to train a feature extractor remained to be a popular solution this year [15–18], following last year's trend [11]. Among them, several new attempts at using pre-trained models have achieved comparatively high scores this year. The 1st [15] and 2nd ranked team [16] both fine-tuned pre-trained models BEATs [19] and EAT [20] using low-rank adaptation (LoRA) [21], which may have prevented the model from overfitting by reducing the number of parameters to train. In addition, instead

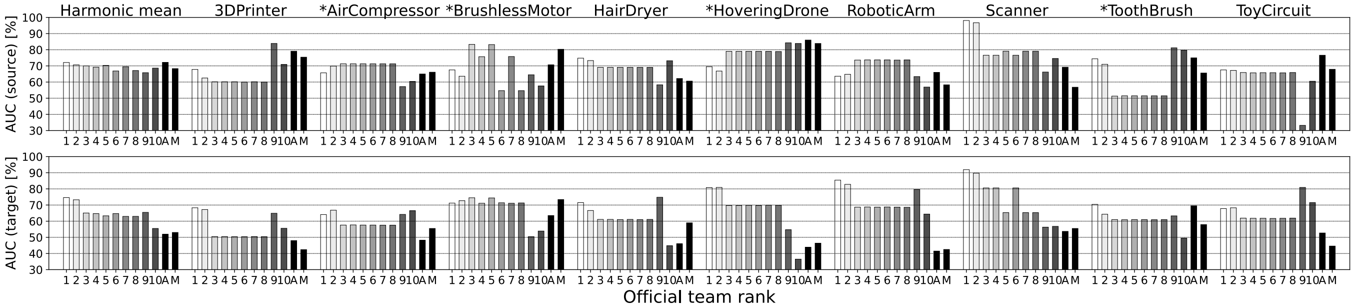


Figure 1: Evaluation results of top 10 teams in ranking. Average source-domain AUC (top) and target-domain AUC (bottom) for each machine type. Labels "A" and "M" on x-axis denote simple Autoencoder mode and selective Mahalanobis mode, respectively. "*" on machine type names indicates that attributes are hidden.

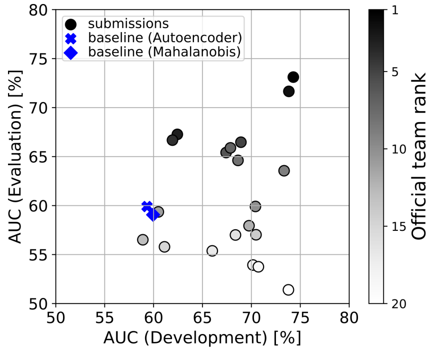


Figure 2: Comparison of average AUC for development and evaluation dataset across teams.

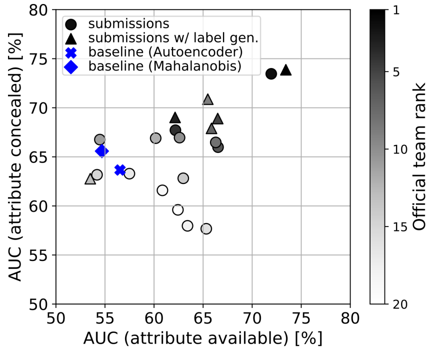


Figure 3: Comparison of average AUC for attribute available and attribute concealed machine types (development and evaluation dataset) across teams. "label gen." refers to pseudo label or auxiliary label generation approaches.

of just ensembling fine-tuned models by adding anomaly scores, both teams created a single model that has two pre-trained models in two branches and fine-tuned them simultaniously. This can automatically balance the influence of the two models on the output. Overall, enhancing ASD performance in this approach could be achieved by investigating deeper into the selection of pre-trained models and the training methodology.

b. Pseudo or auxiliary attribute labeling

Among the classification approaches that proved useful in ASD [11], attribute classification could not be used for nearly half of the machine types this year because attribute information was concealed. To deal with this situation, several teams in the top rankings generated pseudo or auxiliary labels and used them for the auxiliary classification task [15, 22–26]. The 1st ranked team [15] applied agglomerative hierarchical clustering to the audio embeddings, whereas the 9th team [25] proposed and used a bottom-up clustering method to obtain pseudo labels. The 3rd, 5th, and 13th teams [22, 23, 26] also applied clustering methods to spectrograms or statistical features. The 7th team [24] combined the training data with data from the attribute-available machines and used those attribute labels for the classification target.

As shown in Figure 3, teams using these strategies generally had higher AUC values for machines with concealed attributes, indicating the effectiveness of such strategies. However, these strategies mostly worked better only for certain machine types such as Slider or BrushlessMotor, which caused these high average AUC values. This limitation in the effective machine types might be because of the difficulty in distinguishing the sound of the target machine from the background noise only from the audio data. This difficulty can lead to wrongly created pseudo labels based on background noise differences, which does not help models learn the unique features of the target machine. To make this approach more widely effective for various machines, further investigation on how to generate labels, the usable conditions, and what assumptions can be helpful for these strategies is needed.

c. Other novel approaches

Using multiple types of input features such as the log-mel spectrogram and the power spectrum or other features has been introduced by the 4th, 7th, 9th, and several other teams [18, 24, 25, 27]. Several teams carefully selected external datasets and used them for pre-training their model. For example, the 5th and 6th team [17, 23] selected machine-related data or excluded human speech data from AudioSet [28] to make the pre-training data close to the target datasets. The 9th team applied a core-set selection method to AudioSet that selects samples with low anomaly scores as pre-training data, after manually selecting some machine-related classes [25].

5. CONCLUSION

We presented an overview of the task and analysis of the solutions submitted to the DCASE 2024 Challenge Task 2. The task's aim was to develop ASD systems that work for a novel machine type with a single section for each machine type, where the attribute information was concealed for several machine types. We discussed several new approaches that helped improve ASD performance, including ways of using pre-trained models and creating pseudo labels. We hope that all technical reports will contribute to advancements in the academic field and the industrial application of first-shot unsupervised ASD.

6. REFERENCES

[1] Y. Koizumi, S. Saito, H. Uematsu, and N. Harada, "Optimizing acoustic feature extractor for anomalous sound detection based on Neyman-Pearson lemma," in *Proc. EUSIPCO*, 2017, pp. 698–702.

[2] Y. Kawaguchi and T. Endo, "How can we detect anomalies from subsampled audio signals?" in *Proc. IEEE MLSP*, 2017.

[3] Y. Koizumi, S. Saito, H. Uematsu, Y. Kawachi, and N. Harada, "Unsupervised detection of anomalous sound based on deep learning and the Neyman-Pearson lemma," *IEEE/ACM TASLP*, vol. 27, no. 1, pp. 212–224, Jan. 2019.

[4] Y. Kawaguchi, R. Tanabe, T. Endo, K. Ichige, and K. Hamada, "Anomaly detection based on an ensemble of dereverberation and anomalous sound extraction," in *Proc. IEEE ICASSP*, 2019, pp. 865–869.

[5] Y. Koizumi, S. Saito, M. Yamaguchi, S. Murata, and N. Harada, "Batch uniformization for minimizing maximum anomaly score of DNN-based anomaly detection in sounds," in *Proc. IEEE WASPAA*, 2019, pp. 6–10.

[6] K. Suefusa, T. Nishida, H. Purohit, R. Tanabe, T. Endo, and Y. Kawaguchi, "Anomalous sound detection based on interpolation deep neural network," in *Proc. IEEE ICASSP*, 2020, pp. 271–275.

[7] H. Purohit, R. Tanabe, T. Endo, K. Suefusa, Y. Nikaido, and Y. Kawaguchi, "Deep autoencoding GMM-based unsupervised anomaly detection in acoustic signals and its hyperparameter optimization," in *Proc. DCASE Workshop*, 2020, pp. 175–179.

[8] Y. Koizumi, Y. Kawaguchi, K. Imoto, T. Nakamura, Y. Nikaido, R. Tanabe, H. Purohit, K. Suefusa, T. Endo, M. Yasuda, and N. Harada, "Description and discussion on DCASE2020 challenge task2: Unsupervised anomalous sound detection for machine condition monitoring," in *Proc. DCASE Workshop*, 2020, pp. 81–85.

[9] Y. Kawaguchi, K. Imoto, Y. Koizumi, N. Harada, D. Niizumi, K. Dohi, R. Tanabe, H. Purohit, and T. Endo, "Description and discussion on DCASE 2021 challenge task 2: Unsupervised anomalous detection for machine condition monitoring under domain shifted conditions," in *Proc. DCASE Workshop*, 2021, pp. 186–190.

[10] K. Dohi, K. Imoto, N. Harada, D. Niizumi, Y. Koizumi, T. Nishida, H. Purohit, R. Tanabe, T. Endo, M. Yamamoto, and Y. Kawaguchi, "Description and discussion on DCASE 2022 challenge task 2: Unsupervised anomalous sound detection for machine condition monitoring applying domain generalization techniques," in *Proc. DCASE Workshop*, 2022, pp. 26–30.

[11] K. Dohi, K. Imoto, N. Harada, D. Niizumi, Y. Koizumi, T. Nishida, H. Purohit, R. Tanabe, T. Endo, and Y. Kawaguchi, "Description and discussion on DCASE 2023 challenge task 2: First-shot unsupervised anomalous sound detection for machine condition monitoring," in *Proc. DCASE Workshop*, 2023, pp. 31–35.

[12] N. Harada, D. Niizumi, D. Takeuchi, Y. Ohishi, M. Yasuda, and S. Saito, "ToyADMOS2: Another dataset of miniature-machine operating sounds for anomalous sound detection under domain shift conditions," in *Proc. DCASE Workshop*, 2021, pp. 1–5.

[13] K. Dohi, T. Nishida, H. Purohit, R. Tanabe, T. Endo, M. Yamamoto, Y. Nikaido, and Y. Kawaguchi, "MIMII DG: Sound dataset for malfunctioning industrial machine investigation and inspection for domain generalization task," in *Proc. DCASE Workshop*, 2022.

[14] N. Harada, N. Daisuke, T. Daiki, O. Yasunori, and Y. Masahiro, "First-shot anomaly detection for machine condition monitoring: a domain generalization baseline," in *Proc. EUSIPCO*, 2023, pp. 191–195.

[15] Z. Lv, A. Jiang, B. Han, Y. Liang, Y. Qian, X. Chen, J. Liu, and P. Fan, "Aithu system for first-shot unsupervised anomalous sound detection," DCASE2024 Challenge, Tech. Rep., June 2024.

[16] A. Jiang, X. Zheng, Y. Qiu, W. Zhang, B. Chen, P. Fan, W.-Q. Zhang, C. Lu, and J. Liu, "Thuee system for first-shot unsupervised anomalous sound detection," DCASE2024 Challenge, Tech. Rep., June 2024.

[17] Y. Liu, "Dual-mode framework for first-shot unsupervised anomalous sound detection in machine condition monitoring," DCASE2024 Challenge, Tech. Rep., June 2024.

[18] T. Wu, J. wen, Z. Yan, and X. Cheng, "Anomalous sound detection with three-subnetworks and pre-trained models," DCASE2024 Challenge, Tech. Rep., June 2024.

[19] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, Z. Chen, W. Che, X. Yu, and F. Wei, "BEATs: Audio pre-training with acoustic tokenizers," in *Proc. ICML*, 2023, pp. 5178–5193.

[20] W. Chen, Y. Liang, Z. Ma, Z. Zheng, and X. Chen, "Eat: Self-supervised pre-training with efficient audio transformer," *arXiv preprint arXiv:2401.03497*, 2024.

[21] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *Proc. ICLR*, 2021.

[22] R. Zhao, K. Ren, and L. Zou, "Enhanced unsupervised anomalous sound detection using conditional autoencoder for machine condition monitoring," DCASE2024 Challenge, Tech. Rep., June 2024.

[23] L. Wang, M. Cai, J. Pan, T. Gao, and X. Fang, "Two-step anomaly detection: Integrating attribute classification and generative modeling with attribute inference for diverse machine types," DCASE2024 Challenge, Tech. Rep., June 2024.

[24] F. Chu, Y. Zhou, and M. Qian, "Unified anomaly detection for machine condition monitoring: Handling attribute-rich and attribute-free scenarios," DCASE2024 Challenge, Tech. Rep., June 2024.

[25] F. Takuya, I. Kuroyanagi, and T. Toda, "The nu systems for dcase 2024 challenge task 2," DCASE2024 Challenge, Tech. Rep., June 2024.

[26] J. Tian, H. Zhang, S. Zhang, F. Xiao, Q. Zhu, W. Wang, and J. Guan, "Self-supervised anomalous sound detection with statistical clustering and contrastive learning," DCASE2024 Challenge, Tech. Rep., June 2024.

[27] J. Yang, "Adaptive framework for first-shot unsupervised anomalous sound detection in industrial machine monitoring," DCASE2024 Challenge, Tech. Rep., June 2024.

[28] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. IEEE ICASSP*, 2017, pp. 776–780.

A SOUND DESCRIPTION: EXPLORING PROMPT TEMPLATES AND CLASS DESCRIPTIONS TO ENHANCE ZERO-SHOT AUDIO CLASSIFICATION

Michel Olvera, Paraskevas Stamatiadis, Slim Essid

LTCI, Télécom Paris, Institut Polytechnique de Paris, France
{olvera, paraskevas.stamatiadis, slim.essid}@telecom-paris.fr

ABSTRACT

Audio-text models trained via contrastive learning offer a practical approach to perform audio classification through natural language prompts, such as “this is a sound of” followed by category names. In this work, we explore alternative prompt templates for zero-shot audio classification, demonstrating the existence of higher-performing options. First, we find that the formatting of the prompts significantly affects performance so that simply prompting the models with properly formatted class labels performs competitively with optimized prompt templates and even prompt ensembling. Moreover, we look into complementing class labels by audio-centric descriptions. By leveraging large language models, we generate textual descriptions that prioritize acoustic features of sound events to disambiguate between classes, without extensive prompt engineering. We show that prompting with class descriptions leads to state-of-the-art results in zero-shot audio classification across major ambient sound datasets. Remarkably, this method requires no additional training and remains fully zero-shot.

Index Terms— Zero-shot audio classification, audio-text models, contrastive language-audio pretraining, in-context learning

1. INTRODUCTION

Multimodal contrastive pretraining has been used to train multimodal representation models on large amounts of paired data. This approach leverages contrastive learning to align representations across different modalities, promoting a shared embedding space that improves semantic understanding across modalities. Examples include Contrastive Language-Image Pretraining (CLIP) [1], which aligns visual and textual representations, and the more recent Contrastive Language-Audio Pretraining (CLAP), which extends these principles to align audio and textual representations [2, 3, 4, 5].

Following pretraining, CLAP exhibits a well-structured feature space, yielding robust, general-purpose representations well-suited for downstream training. Moreover, it also demonstrates exceptional transferability as evidenced by its impressive zero-shot performance across classification, captioning, retrieval, and generation tasks [3, 6, 7].

Extensive research on CLIP has revealed that classification scores are significantly influenced by alterations in prompt formulation and language nuances. For instance, varying the description of a concept, using synonyms, or modifying the grammatical structure or wording, substantially affects performance outcomes [8, 9, 10]. Besides, prompts offering more context or specificity tend to yield more accurate results [11, 12, 13].

Similarly, CLAP inherits sensitivity to prompting from its contrastive pretraining approach. Yet, the systematic exploration of prompt robustness in CLAP remains limited, despite few works highlighting the sensitivity of classification to prompt variations [14, 15]. These works, primarily conducted on the ESC50 dataset and limited to up to five prompt templates, shed initial light on these variations. However, robustness to prompt changes is likely to vary across different datasets. Addressing this gap, recent efforts have explored alternative approaches, such as prompt tuning strategies and lightweight adapters, to mitigate the reliance on manually engineered prompts [16, 17] with an explicit focus on adapting CLAP to downstream tasks or new domains.

In this work, we propose a tuning-free approach that prompts CLAP models with descriptions of class labels to enhance zero-shot audio classification. While using keywords such as “audio,” “hear,” and “sound” in prompt templates primes the text encoder to focus on audio-related concepts, we hypothesize that enriching prompts with explicit class descriptions can further enhance the model’s ability to clarify the meaning of class labels, particularly in scenarios where labels are ambiguous. Ambiguity stems from both the textual and audio aspects of the data. Textual ambiguity arises from homonyms, where words possess multiple meanings, and from the lack of contextual clues (*e.g.*, ”bat” as both an animal and sports equipment). On the audio side, ambiguity arises from acoustically similar sound categories, such as distinguishing between bird vocalizations (*e.g.*, raven vs. crow calls) and musical instruments (*e.g.*, violin vs. viola). Thus, detailed prompts may clarify sounds heavily reliant on context, and help disambiguate acoustically similar sounds. Such descriptions can also disambiguate abstract sounds such as “white noise” and compensate for knowledge gaps or limited exposure to certain terms. For instance, clarifying “Geiger counter”, as “a detection device that clicks or beeps when detecting radiation” could improve correlations of audio and text features.

To validate our hypothesis, we leverage Large Language Models (LLMs) for their knowledge of sound semantics. Specifically, we used Mistral[1] to describe the acoustic properties of class labels. Our study demonstrates that using audio-centric descriptions of class labels as prompts helps CLAP better ground acoustic features with semantic descriptions, significantly boosting zero-shot classification scores across major environmental sound datasets. Remarkably, our method even outperforms learnable prompt strategies, all without the need for additional training, while remaining entirely zero-shot.

2. METHODOLOGY

We first describe the zero-shot audio classification task, then our adaptive class selection strategy and finally we motivate our LLM-generated class descriptions.

This work was supported by the Audible project, funded by French BPI.

[1] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.1

Class	Base	Context	Ontology
Mandolin	A stringed musical instrument, played with a plectrum, characterized by its small size, high-pitched sound, and distinctive twang.	A stringed musical instrument with a distinctive, twangy sound, often associated with folk or bluegrass music. Typically played by plucking or strumming the strings, producing a bright, melodic tone.	A stringed musical instrument with a distinctive, twangy sound, often used in folk and pop music.
Rail transport	The sound of trains moving on rails, characterized by the clacking of wheels and the rumbling of engines.	The sound of trains moving along rails, characterized by a steady, rhythmic clacking or clicking noise. Often heard in urban or rural areas with rail infrastructure.	The rumbling and clanking sounds produced by trains moving on rails, characterized by their speed and intensity, classified under transportation-related sounds.
Toot	A short, high-pitched sound produced by blowing air through a small opening, often used as a signal or warning.	A short, sharp sound, typically produced by blowing air through a small opening, such as a whistle or a musical instrument.	A short, high-pitched sound produced by a whistle or other musical instrument, often used as a signal or warning.
Stream	A continuous flow of water or other liquid, often characterized by its sound as it flows over rocks or other obstacles.	A continuous flow of water, often heard in natural environments like rivers, lakes, or waterfalls, characterized by the sound of water flowing over rocks or other surfaces.	A continuous flow of sound, often characterized by its rhythmic patterns and timbre, belonging to the category of natural environmental sounds.

Table 1: Example descriptions of randomly sampled class labels from the datasets considered in this work, generated with Mistral-7B [18].

2.1. Zero-shot audio classification

Given a set of target categories C and a query audio sample a, the zero-shot audio classification protocol in CLAP defines the classification problem as a nearest neighbor retrieval task. The predicted category $\hat{c}$ is determined as follows:

$$\hat{c} = \arg\max_{c \in C} \text{sim}(\phi_{\text{A}}(a), \phi_{\text{T}}(c)), \tag{1}$$

where C represents the set of class labels, a denotes the input audio, and ϕ_{A} and ϕ_{T} are the audio and text encoders, respectively. The function $\text{sim}(\cdot,\cdot)$ corresponds to the similarity metric, typically the cosine similarity.

To enhance zero-shot audio classification, we propose using both class labels and their descriptions to resolve ambiguities. Given a set of target categories C, definitions D, the predicted category $\tilde{c}$ is determined by:

$$\tilde{c} = \arg\max_{c \in C} \text{sim}(\phi_{\text{A}}(a), \phi_{\text{T}}(c + d_c))), \tag{2}$$

where $d_c \in D$ is the description corresponding to class c, and the $+$ operator denotes the textual combination of the class label c and its description d_c.

2.2. Adaptive class description selection

We devise an adaptive strategy that incorporates descriptions selectively for classes potentially ambiguous to the text encoder. Let $\text{P}_{\text{class-only}}$ and $\text{P}_{\text{class-description}}$ represent the classification performance for class c using setups involving classes only or classes with descriptions as in Equations (1) and (2), respectively. We decide for class c which setup to apply through the decision function $\text{M}(c)$:

$$\text{M}(c) = \begin{cases} \hat{c} & \text{if } \text{P}_{\text{class-only}} \geq \text{P}_{\text{class-description}} \\ \tilde{c} & \text{if } \text{P}_{\text{class-description}} > \text{P}_{\text{class-only}}. \end{cases} \tag{3}$$

The function $\text{M}(c)$ decides whether a class should include a description based on cross-validation of results.

2.3. Generation of audio-centric descriptions with LLMs

Given audio event class labels, we propose to use Large Language Models (LLMs) to generate audio-centric descriptions for them automatically, as manual collection of descriptions entails a labor-intensive endeavor. LLMs, trained on vast text data, have a deep understanding of language, which we exploit for their knowledge of sound semantics. Our method, adapted from [19], involves three steps. First, we provide a general description of the task. Second, we combine these instructions with in-context demonstrations, including a few paired label-description examples. Finally, we provide the LLM with the class labels, heuristic constraints, and specific output format details to generate audio-centric descriptions.

Using this method, we generated three types of descriptions: *base descriptions*, *context-aware descriptions*, and *ontology-aware descriptions*. All are audio-centric. *Base descriptions* reflect the acoustic properties and characteristic sounds of the class labels. *Context-aware* descriptions add details about the typical locations and circumstances of encountering the sounds, including the physical environment, associated objects, and the function of the sound within its context. *Ontology-aware* descriptions capture the acoustic properties and characteristic sounds of each class label while also considering their relationships with coarse high-level concepts. Table 1 provides a few examples of the generated descriptions. The complete list of class descriptions and the prompts used to generate them are available on our companion website.[2]

3. EXPERIMENTAL SETUP

We detail our experimental approach, including model and dataset selection, evaluation metrics, and experiments to explore different prompt strategies and their impact on classification.

3.1. Models

We adopt two state-of-the-art audio-text models pre-trained via contrastive learning, namely LAION-CLAP (LA) and Microsoft CLAP 2023 (MS). The former utilizes RoBERTa [20] as its text encoder, while the latter leverages GPT-2 [21]. Both models rely on HTS-AT [22] as their audio encoder.

3.2. Datasets and evaluation metrics

Downstream datasets. We select six major environmental sound datasets tailored for either single-class or multi-label classification. These include: **ESC50** [23], which contains 50 environmental sound classes with 2k labeled samples of 5 seconds each; **US8K** [24], comprising 10 urban sound classes and 8k labeled sound excerpts of 4 seconds each; **TUT2017** [25], consisting of 15 acoustic scenes classes and 52k files of 10 seconds each; **FSD50K** [26], featuring 51K audio clips of variable length (from 0.3 to 30 seconds each) curated from Freesound and comprising 200 classes;

[2] https://github.com/tpt-adasp/a-sound-description

AudioSet [27], a large-scale dataset encompassing 527 classes, with over 2 million human-labeled sound clips of 10 seconds from YouTube videos; and **DCASE17-T4** [25], a subset of AudioSet focused on 17 classes related to warning and vehicle sounds, containing 30k audio clips of 10 seconds each.

Evaluation setup and metrics. In our evaluation we consider all available splits (train/val/test) or folds, except for AudioSet, where only the test set was used. Note that some datasets do not allow for a fully zero-shot approach, as some audio files used in the evaluation were part of the pretraining data of the considered frozen CLAP models (*e.g.*, AudioSet and FSD50K). We believe that it is still interesting to analyse the corresponding results, bearing this fact in mind during the discussion. We use accuracy as the metric for single-class classification datasets (ESC50, US8K and TUT2017) and mean Average Precision (mAP) for multi-label classification datasets (FSD50K, AudioSet and DCASE17-T4). For experiments involving class-specific descriptions, a 5-fold cross-validation setting is employed. These folds were constructed on the data considered for evaluation *i.e.,* all splits/folds for all datasets, except for AudioSet where the test set is used. In this approach, training folds are used to derive the mapping M from Equation (3), while test folds are used to assess its generalization. Directly evaluating the mapping without cross-validation would yield overly optimistic results due to overfitting.

3.3. Zero-shot audio classification experiments

Prompting with class labels only We explore zero-shot audio classification using prompts with sanitized class labels (*i.e.*, replacing underscores in original labels with spaces, *e.g.*, *dog_barking* becomes *dog barking*). This is motivated by the fact that in our early experiments we observed that this strategy performs competitively compared to prompting with “This is a sound of”, which has been preferred in the literature [14, 4]. Here, we systematically study the impact of using only class labels as prompts on classification performance. We examine four different formats to construct the start and end of a prompt: uppercase with a period (*e.g., Dog barking.*), uppercase without a period (*e.g., Dog barking*), lowercase with a period (*e.g., dog barking.*), and lowercase without a period (*e.g., dog barking*). The format yielding the highest performance for each model, termed as CLS, was selected as a reference for subsequent experiments involving class descriptions.

Prompting with templates. Inspired from CLIP [1], we explore a set of prompt templates as plausible alternatives to “This is a sound of”, all tailored for the zero-shot audio classification task. We curated a set of 33 distinct prompts, drawing some from prior studies [14, 4, 15]. Our objective is to systematically evaluate the performance of these alternative prompts and their ensemble across multiple datasets. Each prompt follows the format *Template + class label*, *e.g.*, “A sound clip of dog barking.”. We thus analyse the performance of three prompt configurations: PT$_{\text{Baseline}}$: The baseline prompt template “This is a sound of”. PT$_{\text{Best}}$: The most effective prompt template identified among the 33 manually crafted alternatives. PT$_{\text{Ensemble}}$: Ensembling text embeddings from all considered prompt templates. Each prompt template begins with an uppercase letter and concludes with a period.

Prompting with class-specific descriptions. We investigate the impact of combining class labels and their descriptions generated by LLMs. The experimental setups include: CLS: Class label only. CD$_{\text{Base}}$: Audio-centric definitions generated by Mistral. CD$_{\text{Context}}$[3]: Context-aware descriptions. CD$_{\text{Ontology}}$: Ontological information related to the class label. CD$_{\text{Dictionary}}$: Definitions (non audio-centric) sourced from the Cambridge Dictionary of English.[4]

4. RESULTS AND DISCUSSION

In this section, we present and discuss the outcomes of our experiments, shedding light on the impact of various prompting strategies and the role of class descriptions in classification performance.

4.1. Sensitivity to prompt format

In Table 2, we report the average classification results across all evaluation datasets to examine the sensitivity of zero-shot classification performance to subtle variations in the input prompt format. We see surprising differences in performance due to minor alterations such as capitalization and punctuation, consistent with findings in [15]. A recent work on LLM behavior confirm that these seemingly minor changes in prompt format influence the model’s internal representations, leading to distinct transformations within the embedding space that alter the output probability distribution in ways that affect classification performance [28]. We observe that, for both models, prompt variations in punctuation, irrespective of capitalization, significantly affect performance more than variations in capitalization without punctuation. Notably, the performance gap between the most and least effective formats was 5.46% for model LA and 8% for model MS, pointing out how critical it is to select an optimal format to maximize classification scores. Consequently, subsequent experiments adopted the best-performing format for each model.

Prompt format	Model	
	LA	MS
class label (*e.g., dog barking*)	0.5059	0.5256
class label. (*e.g., dog barking.*)	0.5524	**0.5735**
Class label (*e.g., Dog barking*)	0.5110	0.49344
Class label. (*e.g., Dog barking.*)	**0.5605**	0.5395

Table 2: Average model performance scores across all datasets for different input prompt formats.

4.2. Comparison of prompting strategies

In Table 3, top-panel, we show results that assess the impact on classification performance when prompting CLAP models using only the class label and various prompt templates and an ensemble of these prompts. Our findings reveal that using the class label alone (CLS) often yields superior performance compared to the prompt template ”This is a sound of” (PT$_{\text{Baseline}}$). Specifically, CLS demonstrates better results than PT$_{\text{Baseline}}$ on the majority of datasets, with model MS showing an absolute improvement of 1.07%. However, for model LA, CLS showed a slight underperformance of 0.67%, largely due to lower scores on the TUT2017 and DCASE17-T4 datasets.

[3] We did not consider context-aware descriptions for TUT2017 because these were very similar to base descriptions. Unlike other datasets, TUT2017 comprises labels that refer to acoustic scenes. This explains the similarity, as both type of descriptions indicate context.

[4] When definitions where not available in the Cambridge Dictionary, definitions were sourced from WordNet, Wikipedia, and FreeBase.

Method	ESC50		US8K		TUT2017		DCASE17-T4		FSD50K		AudioSet		Average	
	LA	MS	LA	MS	LA	MS	LA	MS	LA	MS	LA	MS	LA	MS
CLS	0.9280	0.9280	0.7980	0.8737	0.4242	0.5717	0.4443	0.3772	0.5409	0.5137	0.2277	0.1764	0.5605	0.5735
PT$_{\text{Baseline}}$	0.915	0.893	0.7747	0.7855	0.4890	0.4547	0.4670	0.4674	0.5308	0.5052	0.2269	0.2708	0.5672	0.5628
PT$_{\text{Best}}$	0.9415	0.9585	0.8133	0.8624	0.5041	0.6192	0.5220	0.4583	0.5765	0.5372	0.2855	0.2708	0.6071	0.6176
PT$_{\text{Ensemble}}$	0.9295	0.95	0.7893	0.8506	0.4944	0.6111	0.4851	0.4075	0.5744	0.5424	0.2560	0.2063	0.5881	0.5946
Adaptive class description selection (mean scores across five folds)														
CD$_{\text{Dictionary}}$	0.9535	0.9205	0.8632	0.8891	0.5770	0.5630	0.4704	0.3776	0.5623	0.4972	0.2727	0.1924	0.6165	0.5733
CD$_{\text{Base}}$	0.9480	0.9505	0.8336	0.8926	0.5790	**0.6219**	0.4705	0.3911	0.5654	0.5039	0.2803	0.1963	0.6128	0.5927
CD$_{\text{Context}}$	0.9455	0.9595	0.8597	0.8782	-	-	**0.4742**	0.3801	**0.5720**	0.5128	**0.2891**	0.2022	**0.6281**	0.5865
CD$_{\text{Ontology}}$	0.9495	**0.9635**	0.8480	**0.9017**	0.5030	0.5670	0.4589	0.3748	0.5676	0.5074	0.2830	0.1998	0.6017	0.5857
CD$_{\text{All}}$	0.9491	0.9485	0.8511	0.8904	0.5530	0.5840	0.4685	0.3809	0.5668	0.5053	0.2813	0.1976	0.6142	0.5845
SOTA	0.96 [15]		0.8526 [17]		0.5438 [17]		-		0.52 [15]		0.102 [17]		-	

Table 3: Zero-shot classification scores across 6 downstream tasks. Evaluation metrics: Accuracy for ESC50, US8K and TUT2017; mean Average Precision (mAP) for DCASE17-T4, FSD50K and AudioSet.

We report the best-performing prompt template, PT$_{\text{Best}}$, among those considered as plausible alternatives to PT$_{\text{Baseline}}$ for each dataset. On average, PT$_{\text{Best}}$ outperformed PT$_{\text{Baseline}}$, with an absolute improvement of 3.99% and 5.48% for LA and MS, respectively. The relevance of this result brings to light the existence of better manually crafted prompt templates than *This is a sound of*. Table 4 lists the best-performing prompt template for each evaluation dataset. Interestingly, the absence of a "universal" template calls for customization to specific datasets and models to optimize performance, given that certain templates may align better with particular dataset labels. Additionally, prompt ensembling (PT$_{\text{Ensemble}}$) outperformed individual prompts like CLS and PT$_{\text{Baseline}}$, but did not exceed PT$_{\text{Best}}$, which can be attributed to less effective prompts in the ensemble, potentially diminishing its overall efficacy.

Dataset	Models	
	LA	MS
ESC50	*Listen to*	*A recording of*
US8K	*I can hear*	*Listen to an audio of*
TUT2017	*This is a sound track of*	*Listen to an audio recording of*
DCASE17-T4	*A sound clip of*	*This is a sound of*
FSD50K	*A sound recording of*	*This is*
AudioSet	*This is an audio clip of*	*This is a sound of*

Table 4: Best-performing prompt templates per dataset.

4.3. Impact of class-specific descriptions

In Table 3, middle-panel, we assess the impact of class-specific descriptions on classification performance through our adaptive selection strategy, which determines which classes benefit from explicit descriptions. Our findings indicate that introducing class descriptions is indeed beneficial for disambiguating difficult classes, with audio-centric descriptions generally outperforming dictionary definitions. Focusing on model LA, class descriptions with contextual information (CD$_{\text{Context}}$) yielded the best results on average. While model MS also benefited from class-specific descriptions, it showed modest gains across datasets, likely due to its pretraining on a larger volume of data, including more audio-caption pairs. For model MS, base audio-centric descriptions of class labels CD$_{\text{Base}}$ were the most effective, but still could not outperform prompt template-based methods in the top-panel for datasets such as DCASE17-T4, FSD50k and AudioSet. However, our adaptive strategy incorporating all types of descriptions (CD$_{\text{All}}$) did not generalize as effectively compared to individual setups, which was somewhat disappointing.

A comparison with state-of-the-art zero-shot audio classification scores reported in the literature, as shown in bottom line of Table 3, reveals that our approach outperforms these benchmarks, including those utilizing prompt tuning strategies such as [17], across all evaluated datasets. The improvements are particularly notable for the US8K, TUT2017, FSD50K, and AudioSet datasets.

4.4. Disambiguation of classes through descriptions

In Table 5, we show the top-3 classes with the greatest absolute improvement in classification using base descriptions compared to the simple use of class labels for the AudioSet and FSD50K datasets. We observe some words are ambiguous in meaning, for which an explicit description is beneficial as indicated by the large absolute improvements. A full list of relative improvements for all datasets is available on our companion website.

Dataset	Class label	Δ Improvement [%]
AudioSet	*Bagpipes*	+40.12
	Fire engine, fire truck (siren)	+39.79
	Gargling	+36.34
FSD50K	*Fowl*	+67.75
	Scratching (performance technique)	+67.21
	Purr	+60.49

Table 5: Top-3 classes with highest absolute improved classification for model MS on AudioSet and FSD50K datasets using base audio-centric descriptions.

5. CONCLUSION

We demonstrated that prompt templates and class-specific descriptions can significantly impact the performance of zero-shot audio classification. While simple class labels can be highly effective, carefully crafted prompt templates and context-aware descriptions offer substantial improvements. Our findings advocate for a nuanced approach to prompt engineering, where the choice of format, content, and contextual information are tailored to the specific requirements of the model and dataset. Future work could explore automated methods for generating optimal prompts and descriptions, to further boost zero-shot audio classification scores.

6. REFERENCES

[1] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *Proc. ICML*. PMLR, 2021, pp. 8748–8763.

[2] A. Guzhov, F. Raue, J. Hees, and A. Dengel, "Audioclip: Extending clip to image, text and audio," in *Proc. ICASSP*. IEEE, 2022, pp. 976–980.

[3] B. Elizalde, S. Deshmukh, and H. Wang, "Natural language supervision for general-purpose audio representations," in *Proc. ICASSP*. IEEE, 2024, pp. 336–340.

[4] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, "Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation," in *Proc. ICASSP*. IEEE, 2023, pp. 1–5.

[5] I. Manco, E. Benetos, E. Quinton, and G. Fazekas, "Contrastive audio-language learning for music," *arXiv preprint arXiv:2208.12208*, 2022.

[6] X. Mei, C. Meng, H. Liu, Q. Kong, T. Ko, C. Zhao, M. D. Plumbley, Y. Zou, and W. Wang, "Wavcaps: A chatgpt-assisted weakly-labelled audio captioning dataset for audio-language multimodal research," *arXiv preprint arXiv:2303.17395*, 2023.

[7] H. Liu, Z. Chen, Y. Yuan, X. Mei, X. Liu, D. Mandic, W. Wang, and M. D. Plumbley, "Audioldm: Text-to-audio generation with latent diffusion models," *arXiv preprint arXiv:2301.12503*, 2023.

[8] M. Yuksekgonul, F. Bianchi, P. Kalluri, D. Jurafsky, and J. Zou, "When and why vision-language models behave like bags-of-words, and what to do about it?" in *Proc. ICLR*, 2023.

[9] B. An, S. Zhu, M.-A. Panaitescu-Liess, C. K. Mummadi, and F. Huang, "Perceptionclip: Visual classification by inferring and conditioning on contexts," in *Proc. ICLR*, 2024.

[10] A. Salinas and F. Morstatter, "The butterfly effect of altering prompts: How small changes and jailbreaks affect large language model performance," *arXiv preprint arXiv:2401.03729*, 2024.

[11] S. Pratt, I. Covert, R. Liu, and A. Farhadi, "What does a platypus look like? generating customized prompts for zero-shot image classification," in *Proc. ICCV*, 2023, pp. 15 691–15 701.

[12] K. Roth, J. M. Kim, A. Koepke, O. Vinyals, C. Schmid, and Z. Akata, "Waffling around for performance: Visual classification with random words and broad concepts," in *In Proc. of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 15 746–15 757.

[13] M. J. Mirza, L. Karlinsky, W. Lin, H. Possegger, M. Kozinski, R. Feris, and H. Bischof, "Lafter: Label-free tuning of zero-shot classifier using language and unlabeled image collections," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[14] B. Elizalde, S. Deshmukh, M. Al Ismail, and H. Wang, "Clap learning audio concepts from natural language supervision," in *Proc. ICASSP*. IEEE, 2023, pp. 1–5.

[15] S. S. Kushwaha and M. Fuentes, "A multimodal prototypical approach for unsupervised sound classification."

[16] Y. Li, X. Wang, and H. Liu, "Audio-free prompt tuning for language-audio models," in *Proc. ICASSP*. IEEE, 2024, pp. 491–495.

[17] S. Deshmukh, R. Singh, and B. Raj, "Domain adaptation for contrastive audio-language models," *arXiv e-prints*, pp. arXiv–2402, 2024.

[18] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, *et al.*, "Mistral 7b," *arXiv preprint arXiv:2310.06825*, 2023.

[19] A.-M. Oncescu, J. F. Henriques, A. Zisserman, S. Albanie, and A. S. Koepke, "A sound approach: Using large language models to generate audio descriptions for egocentric text-audio retrieval," in *Proc. ICASSP*. IEEE, 2024, pp. 7300–7304.

[20] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[21] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[22] K. Chen, X. Du, B. Zhu, Z. Ma, T. Berg-Kirkpatrick, and S. Dubnov, "Hts-at: A hierarchical token-semantic audio transformer for sound classification and detection," in *Proc. ICASSP*. IEEE, 2022, pp. 646–650.

[23] K. J. Piczak, "ESC: Dataset for Environmental Sound Classification," in *Proc. ACM-MM*. ACM Press, pp. 1015–1018. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2733373.2806390

[24] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *Proc. ACM-MM*, 2014, pp. 1041–1044.

[25] A. Mesaros, A. Diment, B. Elizalde, T. Heittola, E. Vincent, B. Raj, and T. Virtanen, "Sound event detection in the dcase 2017 challenge," *Proc. IEEE/ACM Trans. Audio Speech Lang.*, vol. 27, no. 6, pp. 992–1006, 2019.

[26] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "Fsd50k: an open dataset of human-labeled sound events," *IEEE/ACM Trans. Audio Speech Lang.*, vol. 30, pp. 829–852, 2021.

[27] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. ICASSP*. IEEE, 2017, pp. 776–780.

[28] M. Sclar, Y. Choi, Y. Tsvetkov, and A. Suhr, "Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting," in *Proc. ICLR*, 2024.

ESTIMATED AUDIO–CAPTION CORRESPONDENCES IMPROVE LANGUAGE-BASED AUDIO RETRIEVAL

Paul Primus[1], *Florian Schmid*[1], *and Gerhard Widmer*[1,2]

[1]Institute of Computational Perception (CP-JKU)
[2]LIT Artificial Intelligence Lab
Johannes Kepler University, Austria

ABSTRACT

Dual-encoder-based audio retrieval systems are commonly optimized with contrastive learning on a set of matching and mismatching audio–caption pairs. This leads to a shared embedding space in which corresponding items from the two modalities end up close together. Since audio–caption datasets typically only contain matching pairs of recordings and descriptions, it has become common practice to create mismatching pairs by pairing the audio with a caption randomly drawn from the dataset. This is not ideal because the randomly sampled caption could, just by chance, partly or entirely describe the audio recording. However, correspondence information for all possible pairs is costly to annotate and thus typically unavailable; we, therefore, suggest substituting it with *estimated correspondences*. To this end, we propose a two-staged training procedure in which multiple retrieval models are first trained as usual, i.e., without estimated correspondences. In the second stage, the audio–caption correspondences predicted by these models then serve as prediction targets. We evaluate our method on the ClothoV2 and the AudioCaps benchmark and show that it improves retrieval performance, even in a restricting self-distillation setting where a single model generates and then learns from the estimated correspondences. We further show that our method outperforms the current state of the art by 1.6 pp. mAP@10 on the ClothoV2 benchmark.

Index Terms— Language-based Audio Retrieval, Audio–Caption Correspondences

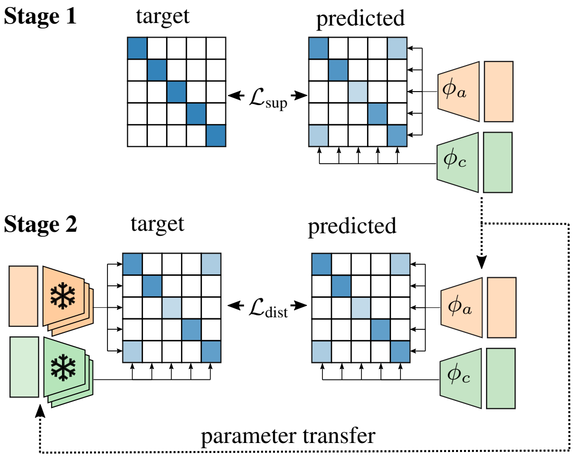


Figure 1: Audio and descriptions are transformed into the shared audio–caption embedding space via the audio and description embedding models ϕ_{a} and ϕ_{c}, respectively. In stage 1, we assume that audio a_i and caption c_j do not match if $i \neq j$ and train the model with contrastive loss $\mathcal{L}_{\text{sup}}$. Stage 2 uses predictions ensembled from several Stage 1 models (bottom left) to estimate the correspondence between a_i and c_j; those estimates then serve as prediction targets instead of the ground truth from stage 1. Stage 2 model parameters are initialized with stage 1 parameters, and the corresponding loss is denoted as $\mathcal{L}_{\text{dist}}$.

1. INTRODUCTION

Language-based audio retrieval systems search for audio recordings based on textual descriptions. Such systems are of practical interest because they allow users to intuitively specify arbitrary acoustic concepts of interest (such as acoustic events, qualities of sound, and temporal relationships) without relying on a predefined set of tags or categories. However, language-based retrieval is difficult from a technical perspective because it requires deriving comparable semantic representations for raw audio signals and sequences of words. Typical audio retrieval systems [1, 2, 3, 4] achieve this via a dual-encoder architecture that projects the textual query and the candidate audio recordings into a shared multi-modal metric space where the audio recordings are then ranked based on their distance to the textual query (for a different approach, see previous work by Labbé et al. [5]).

Previous studies have explored multiple directions to improve language-based audio retrieval systems, such as using better pre-trained embedding models [6], augmentation techniques for both audio and text [7], artificial captions generated with large language models [8, 9, 6], or hybrid content and metadata based retrieval systems [10]. In this work, we expand on the previously proposed idea of utilizing non-binary audio–caption correspondences for training retrieval models [11]. However, instead of relying on crowd-sourced correspondences, our method estimates them via an ensemble of audio retrieval models. To this end, we propose a two-step training procedure that is illustrated in Figure 1. In the following sections, we motivate and describe the proposed two-stage training strategy; we then detail the experimental setup and present results on ClothoV2 [12] and AudioCaps [13]. When trained with large audio–caption datasets, our method outperforms the current state of the art on ClothoV2 by around 1.6 pp. mAP@10. Our submission to the DCASE Challenge 2024 [14], based on the proposed method, took the first rank in task 8. Our implementation, model checkpoints, predictions, and examples are available on GitHub[1].

[1] https://github.com/OptimusPrimus/salsa

2. TEXT-BASED AUDIO RETRIEVAL

Language-based retrieval systems typically consist of two modality encoder networks, one for audio and one for caption embedding, denoted as $\phi_a(\cdot)$ and $\phi_c(\cdot)$, respectively. These encoders learn to embed recordings and descriptions into a shared D-dimensional embedding space such that representations of matching audio snippets and captions are similar. The agreement between audio a_i and description c_j at training or inference time is estimated via the normalized dot product in the multi-modal embedding space:

$$C_{ij} = \frac{\phi_{\text{a}}(a_i)^T \cdot \phi_{\text{c}}(c_j)}{\|\phi_{\text{a}}(a_i)\|^2 \, \|\phi_{\text{c}}(c_j)\|^2}$$

Previous research typically relied on contrastive learning to train audio retrieval models. A usual choice is an adapted version of the Normalized Temperature-scaled cross-entropy (NT-Xnt) loss [15], which converts those agreements into conditional probability distributions over audio snippets and captions via a temperature-scaled softmax operation, where

$$q_a(a_i \mid c_j) = \frac{e^{C_{ij}/\tau}}{\sum_{i=1}^{N} e^{C_{ij}/\tau}}$$

gives the estimated probability that audio a_i corresponds to a given caption c_j, and

$$q_c(c_j \mid a_i) = \frac{e^{C_{ij}/\tau}}{\sum_{j=1}^{N} e^{C_{ij}/\tau}}$$

gives the estimated probability that caption c_j corresponds to a given audio a_i. The training objective is then to minimize the cross-entropy (denoted as H) between the estimated and the actual correspondence probabilities, q and p, respectively.

$$\mathcal{L}_{\text{sup}} = H(p_a, q_a) + H(p_c, q_c)$$

However, the true correspondence probabilities p for audio a_i and caption c_j with $i \neq j$ are not generally available because audio retrieval datasets (e.g., [12, 13, 8]) typically only provide a set of N matching audio and caption pairs $\{(a_i, c_i)\}_{i=1}^{N}$, but no correspondence annotations for the case $i \neq j$. Previous studies thus assumed that c_j does not describe a_i if $i \neq j$, which is reasonable if the dataset holds a large variety of recordings with very specific descriptions. Using this assumption, the target probability distributions p for recordings and captions can then be defined as follows:

$$p_a(a_i \mid c_j) := \mathbb{1}_{i=j} \text{ and } p_c(c_j \mid a_i) := \mathbb{1}_{i=j}$$

Similar to Xie [11], we argue that relying on this assumption is not ideal, mainly for two reasons:

1. It is only valid if each caption in the dataset describes *exactly one* recording, which is not the case for popular audio retrieval datasets such as ClothoV2, AudioCaps, and WavCaps, as demonstrated in [14].
2. Binary correspondences are limited to modeling exact matches between audio recordings and captions. However, we believe that incentivizing the model to place partially matching captions closer to the corresponding audio recording in the multi-modal embedding space is beneficial.

Xie et al. [11] crowdsourced pairwise correspondence scores of audios snippets and captions in a previous study but did not find significant benefits when incorporating binarized versions of those scores during training. We still hypothesize that additional correspondence annotations can provide useful guidance during training; however, there are no large-scale datasets with complete correspondence annotations due to the high cost associated with annotating N^2 audio–caption pairs for large N.

3. PROPOSED METHOD

To obtain audio–caption correspondences without relying on human annotators, we suggest estimating them with an ensemble of M independently pre-trained audio retrieval models. We chose to train those models as described in the previous section; however, other approaches like the method proposed in [5] might lead to comparable results. In our setup, the predicted pairwise agreements are ensembled as follows:

$$\hat{C}_{ij} = \frac{1}{M} \sum_{m=1}^{M} C_{ij}^{m}$$

We use a softmax operation to convert those agreements to an estimate of the true correspondence probabilities of recordings given a caption

$$\hat{p}_a(a_i \mid c_j) := \frac{e^{\hat{C}_{ij}/\tau}}{\sum_{i=1}^{N} e^{\hat{C}_{ij}/\tau}}$$

and an estimate of the true correspondence probabilities of captions given an audio

$$\hat{p}_c(c_j \mid a_i) := \frac{e^{\hat{C}_{ij}/\tau}}{\sum_{j=1}^{N} e^{\hat{C}_{ij}/\tau}}$$

These two probability distributions then serve as prediction targets instead of the deterministic correspondence probabilities p_a and p_c in the NT-Xent loss. We refer to the corresponding loss as distillation loss

$$\mathcal{L}_{\text{dist}} = H(\hat{p}_a, q_a) + H(\hat{p}_c, q_c)$$

due to the conceptual similarity to knowledge distillation [16].

4. EXPERIMENTAL SETUP

4.1. Datasets & Benchmarks

We experimented with two popular audio-retrieval benchmark datasets, namely ClothoV2 [12] and AudioCaps [13]. We additionally use WavCaps [8] for training to compare our method to the current state of the art. We briefly describe the datasets below.

ClothoV2 [12] contains 15-30 second recordings and captions that are between 8 and 20 words long. The provided training, validation, and test split contain 3840, 1045, and 1045 recordings, respectively; each recording is associated with five human-generated captions.

AudioCaps [13] consists of $51,308$ audio recordings taken from AudioSet [17]. Each training and validation recording is associated with one and five human-written captions, respectively. The audio recordings' length is roughly 10 seconds, and the captions are, on average, 9.8 words long.

WavCaps [8] is currently the largest audio–caption dataset available; it contains $403,050$ audio recordings and has been used

audio			ClothoV2				AudioCaps			
embedding	$\hat{p}$	M	mAP@10	R@1	R@5	R@10	mAP@10	R@1	R@5	R@10
PaSST	✗	-	28.93	18.11	43.54	57.57	55.30	40.74	75.89	86.28
PaSST	✓	3	31.25	19.52	46.49	61.30	57.61	42.55	79.04	88.74
PaSST	✓	1	30.18	18.95	45.28	59.43	56.67	41.66	78.00	88.21
ATST	✗	-	28.26	17.36	42.58	56.25	55.86	42.47	74.06	84.26
ATST	✓	3	31.01	19.29	46.33	61.07	59.37	45.43	78.02	88.51
ATST	✓	1	29.83	17.75	43.73	57.81	57.72	43.89	77.17	86.96
MN	✗	-	28.72	17.57	43.73	57.82	55.16	40.26	74.06	87.11
MN	✓	3	30.25	18.49	46.11	59.81	57.06	41.83	77.92	88.45
MN	✓	1	28.96	17.80	44.59	57.84	54.95	39.62	76.66	88.32
hybrid MN [10]	✗	-	29.88	18.39	45.04	58.62	58.61	43.47	79.38	90.16

Table 1: Retrieval performance on the AudioCaps and Clotho benchmarks. Each section corresponds to a different Audio Embedding Model. Results in the first row in each section correspond to results without estimated audio–caption correspondences (i.e., ✗ in column $\hat{p}$). The second row gives results of models fine-tuned with the estimated audio–caption correspondences (i.e., ✓ in column $\hat{p}$ and $M = 3$). The third row gives results in the self-distillation setting (i.e., ✓in column $\hat{p}$ and $M = 1$).

successfully in previous studies to reach state-of-the-art retrieval performance [8, 18, 6]. Each audio file in WavCaps is associated with a synthetic audio caption that was created by instructing the GPT3.5-turbo model to extract relevant sound events from metadata and output a single-sentence description. The generated captions are, on average, 7.8 words long. In order to avoid information leakage between the training and evaluation sets, we excluded the overlapping recordings between WavCaps and the evaluation subsets of ClothoV2.

4.2. Pretrained Embedding Models

We experimented with three audio embedding models (PaSST [19], ATST [20], and MN [21]) and one text embedding model (RoBERTa [22]); below, we briefly describe how we used them for audio and text embedding.

4.2.1. Audio Embedding

PaSST [19] has a positional encoding for inputs of up to 10 seconds; we thus cut the up to 30-second long inputs into non-overlapping 10-second snippets and averaged their embeddings. We used the version of PaSST without patch overlap and applied structured patchout of 2 and 15 patches over frequency and time dimensions, respectively. We used the checkpoint denoted as `passt_s_p16_s16_128_ap468` in our experiments, which is available via GitHub[2].

ATST-Frame [20] (denoted only as ATST in the following) has a positional encoding that is also limited to 10 seconds; we again cut the audio recordings into non-overlapping 10-second snippets and averaged their embeddings to obtain a single embedding vector. During training, we used frequency warping [20] where at most 10% of the higher frequency bins were dropped. We used a publicly available checkpoint of ATST (called `atst_as2M.ckpt`) that was further fine-tuned on the weak labels of AudioSet[3].

EfficientAT MobileNetV3 [21] (referred to as MN in the following) is particularly well suited for experiments with ClothoV2 because the CNN architecture can handle audio recordings of arbitrary length as input. We used the model with ID `mn40_as_ext` in our experiments. The checkpoint is available on GitHub[4].

4.2.2. Sentence Embedding

Roberta large [22] was used for sentence embedding because it gave the best performance in our previous comparison of text embedding models [6]. RoBERTa is a bi-directional self-attention-based sentence encoder that underwent self-supervised pretraining on the BookCorpus [23] and WikiText datasets [24]. The RoBERTa large model has around 354 million parameters.

4.3. Optimization

During pre-training (stage 1), both modality encoders were jointly optimized using gradient descent with a batch size of 64 for PaSST and ATST and 32 for MN. We used the Adam update rule [25] to minimize $\mathcal{L}_{\text{sup}}$ for 20 epochs, with one warmup epoch. Thereafter, the learning rate was decayed from 2×10^{-5} to 10^{-7} using a cosine schedule. The hyperparameters of the optimizer were set to PyTorch's [26] defaults.

Fine-tuning (stage 2) was done by minimizing $\mathcal{L}_{\text{dist}}$. Model parameters in stage 2 were initialized with the parameters from stage 1. The training schedule and learning rate were chosen to be the same as in Stage 1 (however, they might benefit from additional tuning). Audio–caption correspondence estimates were obtained by assembling the similarity scores of all three models ($M = 3$) as described in Section 3. We set τ to a constant value of 0.05 in all our experiments.

We used the benchmarks' validation sets to select checkpoints and report results on the test sets here. Our main evaluation criterion for hyperparameter selection was the mean Average Precision among the top-10 results (mAP@10) which is the metric used for ranking systems in the DCASE Challenge. In the results section, we additionally report the recall among the top-1, top-5, and top-10 retrieved results, which allows more detailed analysis and comparison with additional previous work.

[2] https://github.com/kkoutini/PaSST

[3] https://github.com/Audio-WestlakeU/ATST-SED

[4] https://github.com/fschmid56/EfficientAT

5. RESULTS & DISCUSSION

Table 1 summarizes the retrieval performance of our method on the AudioCaps and ClothoV2 benchmarks. Each section in Table 1 corresponds to one of the three audio embedding models. We chose to experiment without external data first to demonstrate the effectiveness of our method. In Section 5.3, we will then show that our method establishes new state-of-the-art performance on the ClothoV2 benchmark when paired with a large audio–caption dataset.

5.1. Does fine-tuning with estimated correspondences lead to improved retrieval performance?

We first pre-trained the three retrieval models without estimated correspondence and report the results in the first row of each section of Table 1. The resulting models were then fine-tuned using the ensembled audio–caption correspondence estimates of all three retrieval models from stage 1. The results are given in the second row of each section in Table 1. We note a substantial increase across all performance metrics for both ClothoV2 and AudioCaps, which indicates that using estimated correspondences has a positive effect.

We additionally compare the proposed method to our recent hybrid content and metadata-based retrieval system [10], denoted as hybrid MN in Table 1 (last section). We find that using the estimated correspondences leads to similar improvements on both benchmarks, but without relying on additional audio metadata such as descriptive tags for retrieval. We hypothesize that combining these two approaches could lead to further performance gains.

5.2. Ablation Study: Is a diverse ensemble required to achieve improvements with estimated audio–caption correspondences?

In the previously described experiments, we relied on ensembled predictions from three diverse models ($M = 3$) to derive the audio–caption correspondences. We want to understand if the performance improvement is a result of distilling from an ensemble of multiple models or if similar results can be achieved in a self-distillation setting. To this end, we dropped the ensembling of multiple models when deriving the correspondences, i.e., we used the same model to generate and then learn from the estimated correspondences. The results are given in the third row of each section in Table 1. We observe that PaSST and ATST benefitted even in this limiting self-distillation setting. However, we also note that MN's performance did not generally improve over the pretraining performance. We hypothesize that this could be fixed with the additional hyperparameter tuning for the second stage. We further observe that using ensemble predictions led to an additional performance improvement over the self-distillation approach (compare rows two and three in each section). We thus recommend using ensembled predictions to estimate audio–caption correspondences whenever additional models are available.

5.3. Comparison to state-of-the-art systems

Current state-of-the-art audio retrieval systems [6, 18] train on multiple audio–caption datasets to increase their performance. To compare our method to these systems under fair conditions, we also increased the size of the training set. To this end, we combined AudioCaps, ClothoV2, and WavCaps (as done in [7]) and pretrained the three previously introduced systems on the merged dataset. The resulting models were fine-tuned on ClothoV2 by minimizing a linear combination of $\mathcal{L}_{\text{sup}}$ and $\mathcal{L}_{\text{dist}}$. We conducted a grid search over the linear combination's weight, the learning rate, and possible ensemble combinations and selected the best PaSST model on the ClothoV2 validation set.

The first section in Table 2 compares the performance of models before and after fine-tuning on ClothoV2. Stage 1 training on the scaled-up dataset (first row in Table 2) already led to better results than training only on ClothoV2. When this model was fine-tuned on ClothoV2 without the estimated correspondences (second row in Table 2), the mAP@10 improved by around 0.9 pp; when the estimated correspondences were used during fine-tuning (third row in Table 2), the mAP@10 increased even more, namely by around 4.6 pp.

The second section in Table 2 compares our method to current state-of-the-art audio retrieval systems. Our proposed method outperforms last year's best single system submission to the DCASE Challenge (Submission 2 of [27]) by around 1.6 pp. without using text augmentations and synthetic captions. The results also show that our approach achieves a higher recall compared to VAST [18], a vision–audio–text model that was trained on 27 million videos.

		ClothoV2			
method	$\hat{p}$	mAP@10	R@1	R@5	R@10
PaSST (stage 1)	✗	35.46	23.64	51.44	64.98
PaSST (stage 2)	✗	36.33	24.31	52.84	65.63
PaSST (stage 2)	✓	**40.14**	**27.69**	**57.03**	**70.39**
DCASE23 [27]	✗	38.56	26.07	55.27	69.30
VAST [18]	-	-	26.9	53.2	66.1

Table 2: First section: Performance of our method on the ClothoV2 benchmark when models were pre-trained on WavCaps, AudioCaps, and ClothoV2. A ✓ in column $\hat{p}$ indicates that estimated correspondences were used when fine-tuning on ClothoV2 in stage 2. Second section: Performance of current state-of-the-art audio-retrieval models.

6. CONCLUSION

In this work, we have explored the use of estimated audio–caption correspondences to train language-based audio retrieval models. We proposed a two-stage training procedure that first estimates the correspondences and then uses those estimated correspondences for training. We showed that ensemble correspondence estimates lead to improved retrieval performance on both AudioCaps and ClothoV2. We further experimented with using the same model to generate and then learn from the estimated correspondences, which led to improved performance for two out of the three investigated retrieval systems. Finally, we scaled up our approach by combining multiple datasets; the resulting model outperforms the previous state-of-the-art on ClothoV2 by around 1.6 pp. mAP@10.

7. ACKNOWLEDGMENT

The LIT AI Lab is financed by the Federal State of Upper Austria. Gerhard Widmer's work is supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme, grant agreement No 101019375 (Whither Music?). Some parts of the computational results presented in this work have been achieved using the Vienna Scientific Cluster (VSC).

8. REFERENCES

[1] A. S. Koepke, A. Oncescu, J. F. Henriques, Z. Akata, and S. Albanie, “Audio retrieval with natural language queries: A benchmark study,” *IEEE Trans. Multim.*, vol. 25, pp. 2675–2685, 2023.

[2] Y. Xin, D. Yang, and Y. Zou, “Improving text-audio retrieval by text-aware attention pooling and prior matrix revised loss,” in *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing, ICASSP, Rhodes Island, Greece*, 2023.

[3] X. Mei, X. Liu, J. Sun, M. D. Plumbley, and W. Wang, “On metric learning for audio-text cross-modal retrieval,” in *Proc. of the 23rd Annual Conf. of the Int. Speech Communication Association, Interspeech, Incheon, Korea*, 2022.

[4] H. Xie, O. Räsänen, K. Drossos, and T. Virtanen, “Unsupervised audio-caption aligning learns correspondences between individual sound events and textual phrases,” in *Proc. of the IEEE Int. Conf. Acoustic., Speech and Signal Process., ICASSP*, 2022.

[5] E. Labbé, T. Pellegrini, and J. Pinquier, “Killing two birds with one stone: Can an audio captioning system also be used for audio-text retrieval?” in *Proc. of the 8th Workshop on Detection and Classification of Acoustic Scenes and Events, DCASE 2022, Helsinki, Finland*, 2022.

[6] P. Primus, K. Koutini, and G. Widmer, “Advancing natural-language based audio retrieval with PaSST and large audio-caption data sets,” in *Proc. of the 8th Workshop on Detection and Classification of Acoustic Scenes and Events, DCASE, Helsinki, Finland*, 2023.

[7] P. Primus and G. Widmer, “Improving natural-language-based audio retrieval with transfer learning and audio & text augmentations,” in *Proc. of the 7th Workshop on Detection and Classification of Acoustic Scenes and Events, DCASE, Nancy, France*, 2022.

[8] X. Mei, C. Meng, H. Liu, Q. Kong, T. Ko, C. Zhao, M. D. Plumbley, Y. Zou, and W. Wang, “Wavcaps: A ChatGPT-assisted weakly-labelled audio captioning dataset for audio-language multimodal research,” *CoRR*, vol. abs/2303.17395, 2023.

[9] G. Zhu and Z. Duan, “Cacophony: An improved contrastive audio-text model,” *CoRR*, vol. abs/2402.06986, 2024.

[10] P. Primus and G. Widmer, “Fusing audio and metadata embeddings improves language-based audio retrieval,” in *Proc. of the 32nd European Signal Processing Conf., EUSIPCO, Lyon, France*, 2023.

[11] H. Xie, K. Khorrami, O. Räsänen, and T. Virtanen, “Crowdsourcing and evaluating text-based audio retrieval relevances,” in *Proc. of the 8th Workshop on Detection and Classification of Acoustic Scenes and Events, DCASE, Helsinki, Finland*, 2023.

[12] K. Drossos, S. Lipping, and T. Virtanen, “Clotho: an Audio Captioning Dataset,” in *Proc. of the IEEE Int. Conf. Acoustic., Speech and Signal Process., ICASSP, Barcelona, Spain*, 2020.

[13] C. D. Kim, B. Kim, H. Lee, and G. Kim, “AudioCaps: Generating captions for audios in the wild,” in *Proc. of the North American Ch. of the Ass. for Computational Linguistics: Human Language Technologies, NAACL-HLT*, 2019.

[14] P. Primus, , and G. Widmer, “A Knowledge Distillation Approach to Improving Language-Based Audio Retrieval Models,” DCASE2024 Challenge, Tech. Rep., June 2024.

[15] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, “A simple framework for contrastive learning of visual representations,” in *Proc. of the 37nd Int. Conf. on Machine Learning, ICML*, 2020.

[16] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” in *NIPS Deep Learning and Representation Learning Workshop, Montreal, Canada*, 2015.

[17] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, “Audio Set: An ontology and human-labeled dataset for audio events,” in *Proc. IEEE Int. Conf. Acoustic., Speech and Signal Process., ICASSP, New Orleans, LA, USA*, 2017.

[18] S. Chen, H. Li, Q. Wang, Z. Zhao, M. Sun, X. Zhu, and J. Liu, “VAST: A vision-audio-subtitle-text omni-modality foundation model and dataset,” in *Annual Conf. on Neural Information Processing Systems 2023, NeurIPS, New Orleans, LA, USA*, 2023.

[19] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, “Efficient training of audio transformers with patchout,” in *23rd Annual Conf. of the Int. Speech Communication Association, Interspeech, Incheon, Korea*, 2022.

[20] X. Li, N. Shao, and X. Li, “Self-supervised audio teacher-student transformer for both clip-level and frame-level tasks,” *IEEE ACM Trans. Audio Speech Lang. Process.*, 2024.

[21] F. Schmid, K. Koutini, and G. Widmer, “Efficient large-scale audio tagging via transformer-to-CNN knowledge distillation,” in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing ICASSP, Rhodes Island, Greece*, 2023.

[22] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “Roberta: A robustly optimized BERT pretraining approach,” *CoRR*, vol. abs/1907.11692, 2019.

[23] Y. Zhu, R. Kiros, R. S. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, “Aligning books and movies: Towards story-like visual explanations by watching movies and reading books,” in *Proc. of the IEEE Int. Conf. on Computer Vision, ICCV, Santiago, Chile*, 2015.

[24] S. Merity, C. Xiong, J. Bradbury, and R. Socher, “Pointer sentinel mixture models,” in *Proc. of the 5th Int. Conf. on Learning Representations, ICLR, Toulon, France*, 2017.

[25] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proc. of the 3rd Int. Conf. on Learning Representations, ICLR, San Diego, CA, USA*, 2015.

[26] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Z. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “PyTorch: An imperative style, high-performance deep learning library,” in *Proc. of the Annual Conf. on Neural Information Processing Systems, NeuRIPS, Vancouver, Canada*, 2019.

[27] P. Primus, K. Koutini, and G. Widmer, “CP-JKU’s submission to task 6b of the DCASE2023 Challenge: Audio retrieval with PaSST and GPT-augmented captions,” DCASE2023 Challenge, Tech. Rep., June 2023.

SYNTHETIC TRAINING SET GENERATION USING TEXT-TO-AUDIO MODELS FOR ENVIRONMENTAL SOUND CLASSIFICATION

Francesca Ronchini, Luca Comanducci, Fabio Antonacci

Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Italy,
{francesca.ronchini, luca.comanducci, fabio.antonacci}@polimi.it

ABSTRACT

In recent years, text-to-audio models have revolutionized the field of automatic audio generation. This paper investigates their application in generating synthetic datasets for training data-driven models. Specifically, this study analyzes the performance of two environmental sound classification systems trained with data generated from text-to-audio models. We considered three scenarios: a) augmenting the training dataset with data generated by text-to-audio models; b) using a mixed training dataset combining real and synthetic text-driven generated data; and c) using a training dataset composed entirely of synthetic audio. In all cases, the performance of the classification models was tested on real data. Results indicate that text-to-audio models are effective for dataset augmentation, with consistent performance when replacing a subset of the recorded dataset. However, the performance of the audio recognition models drops when relying entirely on generated audio.

***Index Terms*—** Text-to-audio generative models, synthetic dataset, environmental sound classification, data augmentation

1. INTRODUCTION

In the past few years, Text-To-Audio (TTA) models have become the new state-of-the-art for what concerns machine learning-based sound synthesis. TTA models are deep learning generative systems designed to generate audio samples based on textual descriptions, commonly referred to as prompts, which are given as input to the models. Several TTA models have been proposed to generate high-quality, realistic audio samples. Pioneering models include AudioGen [1], an auto-regressive generative model, and AudioLDM [2], based on a latent diffusion model [3]. AudioLDM2 [4], a more sophisticated version of AudioLDM, has been recently proposed. Other TTA systems include Tango [5], Make-an-Audio [6], and Audiobox [7].

Thanks to the high-quality generated audio and ease of use, TTA models have been applied to a wide variety of diverse domains such as augmented and virtual reality [8], foley sound generation [9], among others. The versatility of these models makes them potentially applicable to the task of synthetic dataset generation or data augmentation for deep learning models, particularly in cases where data collection is challenging due to privacy concerns or limited data availability. In fact, one limitation of data-driven approaches is the need for large amounts of labeled training data to reach good performances. Unfortunately, dataset acquisition and labeling are time-consuming and biases-prone procedures [10]. Several studies in the field of sound recognition have shown that augmenting the original dataset with synthetic data during the training phase improves system generalization and enhances performances [11, 12, 13, 14, 15]. The synthetic data considered in previous works were generated using signal-processing-based or audio-mixing tools for synthesizing soundscapes, such as Scaper [11] or Pyroadacoustics [16]. These techniques require the manual tuning of different parameters of the sound generation procedure [11, 16], potentially making the process even more cumbersome and error-prone. The introduction of TTA models could be beneficial as they have the potential to overcome these limitations by allowing the generation of the desired audio content through natural language. However, the literature related to the use of TTA for dataset generation is still limited. In [17], Kroher et al. trained a music genre classifier on a fully artificial music dataset generated with MusicGen [18], a text-to-music generation model. The study focused on 5 music genres and the results show that the classifier effectively generalized features learned from artificial data to real music recordings. In [19], the authors fine-tuned AudioLDM to generate both normal and anomalous sounds, which were included in the training dataset for the anomalous sound detection task. The results indicate that generative sounds are promising to achieve performances comparable to state-of-the-art models.

Motivated by these positive preliminary findings [17, 19], this paper investigates how to leverage TTA models in the field of Environmental Sound Classification (ESC) [20]. ESC refers to the task of classifying environmental sounds that can be presented in an audio clip. In our opinion, ESC is an ideal application area to investigate the possibility of including TTA-generated synthetic data for two reasons: TTA models can generate all the sound types present in most ESC datasets; ESC can be considered between the simplest scenarios among the ones considered by the DCASE community. Therefore, it is naturally the first one to address before tackling more complex tasks.

Concurrently to our work, a similar research study addressed the problem applied to speech modeling and audio recognition [21]; here we specifically focus on the ESC task. We consider two state-of-the-art deep learning models for ESC and analyze how their performances vary when TTA-generated data are included as part of the training dataset according to different methodologies: 1) using TTA to perform data augmentation; 2) using TTA data as the sole source of training data; 3) mixing TTA-generated and real data. Audio samples and the code used for this study are available on GitHub[1].

2. EXPERIMENTAL PROCEDURE

In this section, we briefly introduce the TTA models selected for the dataset generation, the prompt strategies for generating it, and the dataset generation process. Sec. 2.4 briefly introduces the used ESC model architectures.

[1] https://ronfrancesca.github.io/Text-to-Audio-ESC/

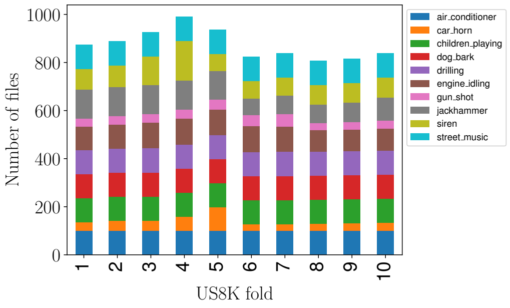


Figure 1: US8K dataset classes distribution per each fold. Colors represent the different sound classes, specified in the legend.

2.1. Text-to-Audio models

We selected two pre-trained models for generating the dataset: AudioGen [1] and AudioLDM2 [4].

AudioGEN is an auto-regressive model that learns a discrete representation of raw audio through an auto-encoding procedure. It then generates audio using a transformer model applied to the learned representation, conditioned on textual features [1].

AudioLDM2 is a continuous latent-diffusion model conditioned via CLAP [22], which removes the need for paired audio-text data during the training process.

2.2. Synthetic dataset generation process

UrbanSound8K (US8K) [23] is the dataset selected for this study. Along with ESC-10 and ESC-50 [24], they serve as the primary datasets used as benchmarks for ESC tasks. The size of US8K makes it more appropriate for training deep learning models compared to ESC-10 and ESC-50, mainly used for evaluation. US8K contains 8732 labeled sounds of 4 s maximum duration of urban sounds from 10 classes. The dataset is divided into 10 folds, used for leave-one-out cross-validation at evaluation time. Fig. 1 reports the sound classes and their distribution in folders. We generated four versions of the US8K dataset: two with AudioLDM2 and two with AudioGen. For each of them, we first generated the total amount of data and then randomly divided it into 10 folds, following the same distribution of US8K.

2.3. Prompt templates

We generated the dataset using two prompt strategies. Both strategies employ a single-instruction sentence containing a sound generation instruction in an urban context, specifying the desired audio class. We explored various templates and informally listened to the generated audio to determine which prompt was the most effective. The first strategy's template is: *"A clear sound of a <class_to_generate> in an urban context."* The template and the adjective choice are based on suggestions from the AudioLDM2 authors' guidelines[2] and studies related to prompt tuning for sound classification [25]. Notably, the use of the adjective *clear* is supported by frequency counts of training data and by its use in other studies [26]. The second strategy uses ChatGPT 3.5 Large Language Model (LLM), which is asked to generate a single sentence to be used as input for a TTA model. Different studies have shown that using an LLM provides diverse and contextually rich prompts [27, 28]. The prompt template suggested by the LLM was: *"Generate a realistic audio representation of the sound of a <class_to_generate> in an urban environment"*. For AudioLDM2 we also used *"Low quality"* as a negative prompt, following the authors' guidelines and implementations in other domains [29]. Depending on the sound class, the templates were adapted to include repetitive sounds for consistency with the study's padding strategy (e.g., dog bark or car horn) or to better specify a sound that might confuse its generation (e.g., siren). The same templates were used for AudioGen to ensure consistency. However, AudioGen does not involve the use of a negative prompt. We are conscious of the fact that handcrafted prompts proposed to generate the data could be a limitation of the current study [30]. Alternative prompt strategies will be considered in future works.

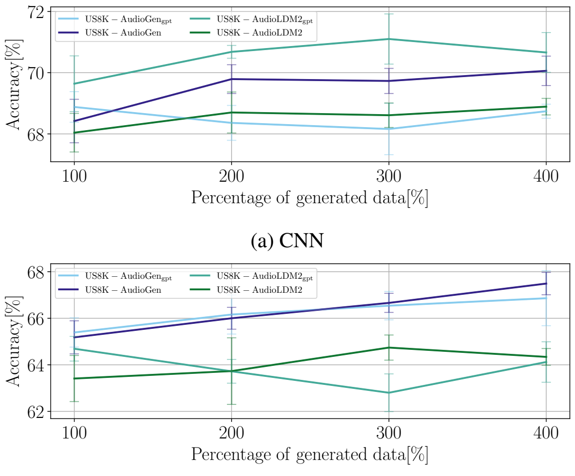
Table 1: Data augmentation comparison between signal-processing-based and TTA-based strategies.

Data aug. method	Accuracy (CNN)	Accuracy (CRNN)
US8K-PS	66.49 (0.60)	65.01 (0.95)
US8K-TS	64.14 (0.80)	62.63 (1.80)
US8K-AudioGen	68.42 (0.71)	65.18 (0.87)
US8K-AudioGen$_{\text{gpt}}$	68.88 (0.50)	**65.39 (0.63)**
US8K-AudioLDM2	68.04 (0.63)	63.41 (0.99)
US8K-AudioLDM2$_{\text{gpt}}$	**69.64 (0.91)**	64.69 (0.53)
US8K (Baseline)	64.68 (0.82)	62.70 (0.65)

2.4. Model Architectures

We purposely select two simple, yet still relevant, architectures for ESC classification since our objective is to focus as much as possible on the quality of the data and not on the complexity of the architectures. Specifically, we considered a Convolutional Neural Network (CNN) and a Convolutional Recurrent Neural Network (CRNN) as ESC models. The CNN is implemented following a similar structure as the one presented in [31]. It is composed of three convolutional layers, each followed by a max-pooling operation, except the last layer. The kernel size, max pooling operation, and dropout parameters are the same as [31]. The CRNN is inspired by [32]. It is composed of seven convolutional blocks followed by a bidirectional GRU layer and a dense layer that generates the final output. We used the same parameters and configuration proposed in [32]. For consistency, the input of both networks consists of TF patches of 3 s taken from the log mel-spectrogram computed from the audio input, as in [31]. All the sounds of US8K have been resampled to 16 kHz, being this the frequency at which the selected TTA models generate sounds. We computed the STFT considering a Hann window of 1024 samples, and 2048 frequency points. We used 64 mel-bands for the log mel-spectrogram with a frequency range between 0 Hz and 8000 Hz. Both networks have been trained for 100 epochs, with batch size of 128 and an early stop condition with patience on the validation loss of 15 epochs. We considered Adam optimizer with a learning rate of 0.001. Samples shorter than 4 s have been padded by repeating the sample until reaching the desired time length. Our implementation of the networks is slightly different than the originals so, as is common in practice, results will not be exactly the same as the original paper.

[2] https://huggingface.co/docs/diffusers/main/en/api/pipelines/audioldm2



(a) CNN

(b) CRNN

Figure 2: Classification accuracy when varying the size of the TTA-generated augmentation dataset. Error bars represent 95% confidence intervals over 5 runs of the experiment.

3. EXPERIMENTS AND RESULTS

This section describes the experiments designed to understand to what extent the TTA-generated dataset impacts the performance of ESC learning-based models. All results are averaged over 5 different runs of the whole 9-fold cross-validation. When referring to the baseline, we mean the ESC models trained with the original version of the US8K dataset.

3.1. Can TTA-augmented datasets increase the accuracy of ESC models?

This first experiment aims to understand how the integration of TTA-generated audio samples as a data augmentation technique affects the accuracy of the considered ESC systems. Data augmentation is a common technique used in ESC tasks to increase and diversify the dataset and improve the performance. Different techniques have been proposed in previous years [31, 33] using signal processing-based methods.

To investigate this, we trained the two networks with the original US8K dataset by augmenting it with one version of the TTA-generated datasets. We also compared the results with two signal processing data augmentation techniques: Time Stretching (TS), which is the process of changing the speed of an audio signal without affecting its pitch, and Pitch Shifting (PS), which is the process of changing the pitch without affecting the speed of the audio sample. While it would have been possible to compare with several augmentation techniques, we chose TS and PS since they are well established in the literature [33]. For this study, we considered the same range of values of PS1 and TS in [31]. In all experiments, for each file, we randomly select only one PS and TS value between the four values proposed in [31] to double the USK8 size. Table 1 reports the accuracy results for the data augmentation techniques considered. *USK-PS* and *USK-TS* stand for PS and TS applied to the US8K dataset, respectively. The last row indicates the accuracy of the ESC systems trained with only the original US8K dataset. The results show that almost all the TTA-based augmentation techniques reach higher performances compared to the signal processing ones. For both models, the best accuracy scores are reached when GPT-based datasets are considered as data augmentation. The CNN model yields its optimal performance when augmented with *AudioLDM2$_{gpt}$*, achieving nearly a 5% increase in accuracy over the baseline. For the CRNN model, the best results are obtained using *AudioGen$_{gpt}$*, reaching a 3% enhancement compared to the baseline. Signal processing data augmentation techniques consistently yield inferior or comparable performances. These findings suggest that incorporating TTA-generated audio samples as a data augmentation technique enhances the performance of the ESC system.

Table 2: Models accuracy when trained only with synthetic data.

Training dataset	Accuracy (CNN)	Accuracy (CRNN)
AudioGen	40.32 (0.29)	38.79 (1.24)
AudioGen$_{gpt}$	**46.04 (0.71)**	**43.96 (1.36)**
AudioLDM2	38.81 (0.56)	36.11 (1.11)
AudioLDM2$_{gpt}$	38.49 (1.21)	32.86 (1.01)
US8K (Baseline)	64.68 (0.82)	62.70 (0.65)

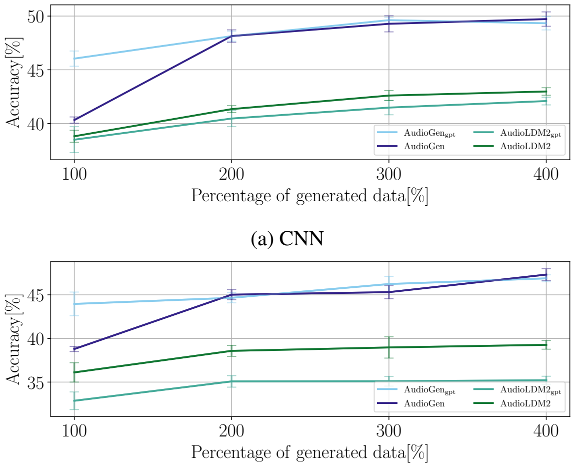
Motivated by these results, we perform a further experiment to understand if increasing the size of the TTA-generated dataset leads to a corresponding increase in performance. We consecutively double the size of the data used for augmentation, up to 400% the original size. We increased the size of the dataset following the same distribution of US8K. Results are reported in Fig. 2, where 100% corresponds to the previous experiment.

Although the CRNN shows improved performance when the original dataset is augmented by 200% to 300% with data from one of the two AudioGen-generated versions, no clear trend is observed for either model. These results suggest that using TTA models for data augmentation is not trivial and requires further investigation.

3.2. Can we rely on only TTA-generated data to train an ESC system?

Motivated by previous results, we explore if TTA-generated data alone can effectively train an ESC system. We trained the ESC models with the different TTA-generated versions of the dataset and tested the models on real data. Table 2 reports the accuracy for the different cases compared with the baseline. The baseline achieves the best performance. However, it is worth noticing that both ESC models (when trained with synthetic data) achieve their highest accuracy when using AudioGen$_{gpt}$ dataset. This emphasizes the preference for AudioGen as a TTA model as a dataset generator for ESC. In contrast, using AudioLDM2 results in inferior performance. However, the results suggest that depending solely on TTA-generated datasets is not yet feasible. Our intuition is that domain adaptation between the TTA-generated used for training and real data used for testing impacts the performances and this will be explored in future investigations.

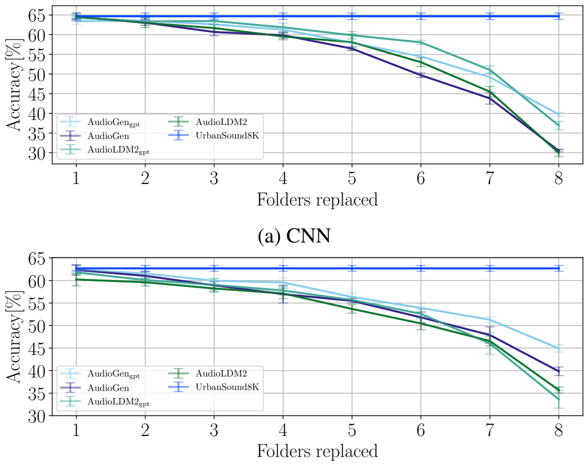
As for the previous experiment, we analyzed if the threshold for achieving baseline performance might be influenced by the quantity of data used at training. Also in this case, we incrementally doubled the dataset size to train models with up to 400% of synthetic data. As reported in Fig. 3, increasing the number of audio data is useful up to 2-3 times the original dataset size, confirming the previous case experiment. Also in this case, both networks achieve higher performances when trained with AudioGen dataset versions, suggesting that AudioGen has the capabilities of generating more



(a) CNN

(b) CRNN

Figure 3: Classification accuracy when varying the size of the training dataset composed of only TTA-generated data. Error bars: 95% confidence intervals over 5 experiment repetitions.



(a) CNN

(b) CRNN

Figure 4: Classification accuracy when incrementally replacing US8K folders using TTA-generated data. Error bars: 95% confidence intervals over 5 experiment repetitions.

realistic audio data that can be used to train ESC models that will then be able to better generalize to real-world data.

3.3. To what extent real data can be safely replaced by synthetic data generated through TTA models?

Previous findings showed that datasets generated through TTA models can enhance performance when used for data augmentation, but solely using synthetic data is not sufficient for effectively training an ESC model. These observations make us wonder if there is a threshold at which synthetic data can effectively replace real data, allowing an ESC model to achieve baseline or better performance while requiring less real data. To investigate this, we conducted several experiments where we incrementally replaced one or more folders of the real US8K dataset with corresponding TTA-generated synthetic folders. Starting with replacing one folder and progressing up to eight folders, we ensured that at least one folder of real data was always included in the training dataset. The folder to be replaced was randomly selected for each iteration of the single experiment. Results are reported in Fig. 4. The straight line indicates the baseline performance. Both ESC models have a similar trend: with up to nearly 20% of real data replaced by synthetic data, the performance is comparable and slightly better for the CRNN. However, beyond this point and up to almost 50% replacement, the accuracy begins to decrease, losing nearly 10%. A noticeable drop in performance occurs beyond the 50% replacement level, with the decline becoming steeper as more real audio files are replaced, ultimately reaching a performance level similar to the experiment described in Section 3.2 when 8 out of 9 folders are replaced. It is worth noting that the AudioGen$_{gpt}$ version of the dataset allows the model to maintain comparable performance even when about 40% of the data is synthetically generated.

4. DISCUSSION

The results show that when training ESC models, TTA-generated data are useful when used to augment or replace part of the real dataset, but they are not ready to completely replace it. While a complete analysis of the reason behind this is out of the scope of this paper and would require further investigations, we report here a few anecdotal causes that we encountered. TTA models do not generate audio related to all the classes with the same effectiveness. For example, a dog barking is better reproduced compared to an audio clip of street music; hammer and air conditioning sounds might be too similar, etc. This is probably part of the reason why the performance drastically drops when only generated data are used during training. We also conducted a preliminary experiment by removing the street music class from the dataset (both training and evaluation), which is the most problematic class. However, no better results were obtained.

Interestingly, the results of this study are in line with the outcome of a parallel study that came out at the time of writing [21], where it is reported a consistent drop in performances when using only synthetic data for similar tasks.

5. CONCLUSIONS AND FUTURE WORKS

This paper investigates the impact of incorporating Text-To-Audio-generated datasets into the training process of ESC systems. We conducted various experiments to explore different methods of integrating and replacing the original dataset with additional training data generated with TTA models. The results show that generated datasets are beneficial when used as data augmentation techniques, but are not ready to be used as the only source of data during training. When replacing part of the real dataset with synthetically generated data, the results are comparable to the baseline up to 10-20% of the data, depending on the ESC model and TTA used. We believe that the obtained results motivate further investigations on the topic. In fact, as the quality of TTAs increases, it is likely that such a training set synthesis approach will be more and more beneficial. Future works will include the exploration of more advanced prompt engineering strategies and the investigation of fine-tuning methods to improve the generation capabilities of TTA models.

6. REFERENCES

[1] F. Kreuk, G. Synnaeve, A. Polyak, U. Singer, A. Défossez, J. Copet, D. Parikh, Y. Taigman, and Y. Adi, "Audiogen: Textually guided audio generation," in *ICLR*, 2022.

[2] H. Liu, Z. Chen, Y. Yuan, X. Mei, X. Liu, D. Mandic, W. Wang, and M. D. Plumbley, "Audioldm: Text-to-audio generation with latent diffusion models," *arXiv:2301.12503*, 2023.

[3] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proc. IEEE/CVF CVPR*, 2022.

[4] H. Liu, Q. Tian, Y. Yuan, X. Liu, X. Mei, Q. Kong, Y. Wang, W. Wang, Y. Wang, and M. D. Plumbley, "Audioldm 2: Learning holistic audio generation with self-supervised pretraining," *arXiv:2308.05734*, 2023.

[5] D. Ghosal, N. Majumder, A. Mehrish, and S. Poria, "Text-to-audio generation using instruction-tuned llm and latent diffusion model," *arXiv:2304.13731*, 2023.

[6] R. Huang, J. Huang, D. Yang, Y. Ren, L. Liu, M. Li, Z. Ye, J. Liu, X. Yin, and Z. Zhao, "Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models," in *ICML*. PMLR, 2023.

[7] A. Vyas, B. Shi, M. Le, A. Tjandra, Y.-C. Wu, B. Guo, J. Zhang, X. Zhang, R. Adkins, W. Ngan, *et al.*, "Audiobox: Unified audio generation with natural language prompts," *arXiv:2312.15821*, 2023.

[8] R. Nordahl, L. Turchet, and S. Serafin, "Sound synthesis and evaluation of interactive footsteps and environmental sounds rendering for virtual reality applications," *Trans. Vis. Comput. Graph.*, 2011.

[9] Y. Chung, J. Lee, and J. Nam, "T-foley: A controllable waveform-domain diffusion model for temporal-event-guided foley sound synthesis," *arXiv:2401.09294*, 2024.

[10] F. Ronchini, R. Serizel, N. Turpault, and S. Cornell, "The impact of non-target events in synthetic soundscapes for sound event detection," in *DCASE Workshop*, 2021.

[11] J. Salamon, D. MacConnell, M. Cartwright, P. Li, and J. P. Bello, "Scaper: A library for soundscape synthesis and augmentation," in *Proc. WASPAA*. IEEE, 2017.

[12] F. Ronchini and R. Serizel, "A benchmark of state-of-the-art sound event detection systems evaluated on synthetic soundscapes," in *Proc. ICASSP*. IEEE, 2022.

[13] F. Gontier, V. Lostanlen, M. Lagrange, N. Fortin, C. Lavandier, and J.-F. Petiot, "Polyphonic training set synthesis improves self-supervised urban sound classification," *The Journal of the Acoustical Society of America*, 2021.

[14] S. Damiano, L. Bondi, S. Ghaffarzadegan, A. Guntoro, and T. van Waterschoot, "Can synthetic data boost the training of deep acoustic vehicle counting networks?" in *Proc. ICASSP*, 2024.

[15] K. M. Ibrahim, A. Perzo, and S. Leglaive, "Towards improving speech emotion recognition using synthetic data augmentation from emotion conversion," in *Proc. ICASSP*, 2024.

[16] S. Damiano and T. van Waterschoot, "Pyroadacoustics: a road acoustics simulator based on variable length delay lines," in *Proc. 25th Int. Conf. Digital Audio Effects*, 2022.

[17] N. Kroher, H. Cuesta, and A. Pikrakis, "Can musicgen create training data for mir tasks?" *arXiv:2311.09094*, 2023.

[18] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, "Simple and controllable music generation," *NeurIPS*, 2024.

[19] H. Zhang, Q. Zhu, J. Guan, H. Liu, F. Xiao, J. Tian, X. Mei, X. Liu, and W. Wang, "First-shot unsupervised anomalous sound detection with unknown anomalies estimated by metadata-assisted audio generation," *arXiv:2310.14173*, 2023.

[20] A. Bansal and N. K. Garg, "Environmental sound classification: A descriptive review of the literature," *ISWA*, 2022.

[21] T. Feng, D. Dimitriadis, and S. Narayanan, "Can synthetic audio from generative foundation models assist audio recognition and speech modeling?" *arXiv e-prints*, 2024.

[22] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, "Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation," in *Proc. ICASSP*. IEEE, 2023.

[23] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *Proc. ACM Multimed.*, 2014.

[24] K. J. Piczak, "ESC: Dataset for environmental sound classification," in *Proc. ACM Multimed.*, 2015.

[25] B. Elizalde, S. Deshmukh, M. Al Ismail, and H. Wang, "Clap learning audio concepts from natural language supervision," in *Proc. ICASSP*. IEEE, 2023.

[26] S. Deshmukh, D. Alharthi, B. Elizalde, H. Gamper, M. A. Ismail, R. Singh, B. Raj, and H. Wang, "Pam: Prompting audio-language models for audio quality assessment," *arXiv preprint arXiv:2402.00282*, 2024.

[27] R. He, S. Sun, X. Yu, C. Xue, W. Zhang, P. Torr, S. Bai, and X. Qi, "Is synthetic data from generative models ready for image recognition?" *arXiv preprint arXiv:2210.07574*, 2022.

[28] X. Mei, C. Meng, H. Liu, Q. Kong, T. Ko, C. Zhao, M. D. Plumbley, Y. Zou, and W. Wang, "Wavcaps: A chatgpt-assisted weakly-labelled audio captioning dataset for audio-language multimodal research," *IEEE/ACM Trans. Audio Speech Lang. Process.*, 2024.

[29] R. Voetman, A. van Meekeren, M. Aghaei, and K. Dijkstra, "Using diffusion models for dataset generation: Prompt engineering vs. fine-tuning," in *Proc. CAIP*. Springer, 2023.

[30] S. Deshmukh, R. Singh, and B. Raj, "Domain adaptation for contrastive audio-language models," *arXiv preprint arXiv:2402.09585*, 2024.

[31] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Process. Lett.*, 2017.

[32] C. Castorena, M. Cobos, J. Lopez-Ballester, and F. J. Ferri, "A safety-oriented framework for sound event detection in driving scenarios," *Applied Acoustics*, 2023.

[33] O. O. Abayomi-Alli, R. Damaševičius, A. Qazi, M. Adedoyin-Olowe, and S. Misra, "Data augmentation and deep learning methods in sound classification: A systematic review," *Electronics*, 2022.

PREDICTION OF *PLEASANTNESS* AND *EVENTFULNESS* PERCEPTUAL SOUND QUALITIES IN URBAN SOUNDSCAPES

Amaia Sagasti, Martín Rocamora, Frederic Font

Music Technology Group
Universitat Pompeu Fabra, Barcelona
name.surname@upf.edu

ABSTRACT

The acoustic environment induces emotions in human listeners. To describe such emotions, ISO-12913 defines *pleasantness* and *eventfulness* as orthogonal properties that characterise urban soundscapes. In this paper, we study different approaches for automatically estimating these two perceptual sound qualities. We emphasize the comparison of three sets of audio features: a first set from the acoustic and psychoacoustic domain, suggested in ISO-12913; a second set of features from the machine listening domain based on traditional signal processing algorithms; and a third set consisting of audio embeddings generated with a pre-trained audio-language deep-learning model. Each feature set is tested on its own and in combination with ground-truth labels about the sound sources present in the recordings to determine if this additional information improves the prediction accuracy. Our findings indicate that the deep-learning representation yields slightly better performance than the other feature sets when predicting pleasantness, but all of them yield similar performance when predicting eventfulness. Nevertheless, deep-learning embeddings present other advantages, such as faster calculation times and greater robustness against changes in sensor calibration, making them more effective for real-time acoustic monitoring. Furthermore, we observe a clear correlation between the sound sources that are present in the urban soundscape and its induced emotions, specially regarding the sensation of pleasantness. Models like the ones proposed in this paper allow for an assessment of the acoustic environment that goes beyond a characterisation solely based on sound pressure level measurements and could be integrated into current acoustic monitoring solutions to enhance the understanding from the perspective of the induced emotions.

Index Terms— Urban soundscapes, acoustic monitoring, emotions, machine-learning, perception

1. INTRODUCTION

Environmental noise regulations are primarily based on sound pressure level (SPL) measurements. For example, the current European Environmental Noise Directive proposes several SPL-based metrics (like *Ld*, *Le*, *Ln* and their combination, *Lden*) to determine permitted noise levels [1]. The limit values depend on factors such as the time of the day and the designated noise sensitivity of the evaluated area. However, other perspectives argue that SPL is insufficient to reliably characterise the acoustic environment [2]. Some psychoacoustic parameters such as loudness and sharpness [3], or episodic memory and visual perception [4], also play a role in shaping the perception of an acoustic environment. In this field of research, the concept of *soundscape* is key, defining the perceptual and emotional construct related to a physical phenomenon (the acoustic environment). The study of soundscapes constitutes a big challenge due to the intrinsic nature of emotions: they are triggered, brief and unconscious [5]. Addressing these difficulties, the ISO-12913 [6, 7, 8] determines a framework to enable international consensus on the definition and conceptual foundation of soundscapes. The standard proposes a model with *pleasantness* and *eventfulness* as main orthogonal axes to characterise soundscape emotional responses, based on the evidence that physiological responses to all types of stimuli can be organized along the dimensions of *valence* and *arousal* [9, 10], or *pleasantness* and *eventfulness* when applied to soundscapes [11].

In this study, we focus on exploring different approaches for automatically estimating the two aforementioned perceptual sound qualities in urban soundscapes. We put emphasis on the comparison of three feature sets for sound representation: the acoustic and psychoacoustic features suggested in ISO-12913, a set of features from the machine listening domain based on traditional signal processing algorithms, and a third set consisting of the audio embeddings generated by a pre-trained language-audio deep-learning model. Each feature set is tested independently and in combination with ground-truth labels about the sound sources present in each recording to determine if this additional information improves the prediction accuracy. Additionally, we examine the models' suitability for real-time acoustic monitoring applications. Our findings indicate that the deep-learning representation yields slightly better performance than the other feature sets when predicting pleasantness, but all of them yield similar results when predicting eventfulness. Nevertheless, deep-learning embeddings present other advantages, such as presumably faster calculation times and greater robustness against changes in sensor calibration, making them more effective for real-time acoustic monitoring. Furthermore, the addition of sound source information improves the prediction accuracy, especially regarding the sensation of pleasantness, indicating a clear correlation between the sound sources present in the urban soundscape and its induced emotions. Models like the ones proposed in this paper allow for an assessment of the acoustic environment that goes beyond a characterisation solely based on SPL measurements and could thereby contribute to the development of more accurate acoustic monitoring techniques, enhancing the understanding of the evaluated environment from an emotional perspective.

The rest of the paper is structured as follows: Section 2 introduces the related work. Section 3 describes the methods used, detailing the dataset and the features employed. Section 4 describes the evaluation process and Section 5 presents the results of our analysis. Finally, Section 6 consists of a discussion of the findings and their implications, followed by a conclusion in Section 7.

2. RELATED WORK

In recent years, many studies have focused on the two-dimensional model for soundscape emotion assessment, resulting in the creation of datasets and the experimenting of algorithms on them. Fan et al.[12] present diverse valence/arousal classifications using their own dataset EMO-SOUNDSCAPES [13, 14]. In an analogous way, ATHUS (Athens Urban Soundscape) [15], created by the authors of [16], is a dataset for urban soundscape quality recognition which includes pleasantness and unpleasantness annotations. Similarly, the ARAUS (Affective Responses to Augmented Urban Soundscapes) dataset [17], combines real urban soundscape recordings with different *audio maskers* including *traffic*, *construction*, *water*, *wind*, *bird*, and *silence*, creating a large-scale dataset of *augmented soundscapes* labelled with pleasantness and eventfulness scores obtained from listening tests developed according to the ISO-12913 [18]. Using psychoacoustic features, the authors run preliminary experiments for the estimation of pleasantness.

Existing research on automatic sound classification provides insights which are also useful for addressing soundscape quality assessment. As an example of early work, Salamon et al. [19] present a set of classification experiments using traditional machine-learning algorithms applied to their own developed urban soundscape datasets [20, 21]. Later sound classification works adopted deep neural networks to address more complex classification problems (e.g., [22]). However, the most recent approaches involve the use of large pre-trained models to extract audio embeddings (i.e. representations) that can be used to address different classification problems and other sound-related tasks such as sound similarity [23]. In particular, Contrastive Language-Audio Pretraining (CLAP) models [24, 25, 26, 27] use contrastive learning to bring audio and text descriptions into a joint multimodal space, and generate sound representations that capture semantically representative information from the audio.

The studies above provide a good framework for research on urban soundscape characterisation. Nevertheless, two important aspects remain unexplored. Firstly, despite existing research showing that the sound sources present in an acoustic environment contribute to its perceived qualities (e.g. natural sounds contribute positively to the pleasantness of an acoustic environment while construction or traffic noise contributes negatively [11, 18]), there is a lack of experiments incorporating such information as an input for automatically characterising soundscapes. Secondly, none of the studies validates the suitability and robustness of the models in real-time contexts, which is essential for the eventual incorporation of the emotional dimension into acoustic monitoring techniques.

3. METHODS

The core methodology for studying different approaches for predicting the perceptual qualities of pleasantness and eventfulness in urban soundscapes involves data selection, feature extraction, and model training. Our main objective is to evaluate the performance of three different feature sets, and determine which one delivers the best results in terms of accuracy and suitability for real-time applications.

3.1. Dataset

We choose the ARAUS dataset for our experiments because it is the most comprehensive available dataset with pleasantness and eventfulness annotations. ARAUS consists of a set of 25,440 unique and 30s-length augmented audios, created by digitally adding audio maskers (see Section 2) to real urban soundscape recordings. They are organised in a five-fold cross-validation set and an independent test set. Based on the soundscape study methodology suggested in the ISO-12913, the audio clips are individually labelled with 1-5 ratings on how *pleasant*, *annoying*, *eventful*, *uneventful*, *vibrant*, *monotonous*, *chaotic* and *calm* they are according to the participants of a listening test. From these ratings, a global value of pleasantness and eventfulness per recording can be calculated as defined in the standard. These values range from -1 to 1, where negative values indicate unpleasantness or uneventfulness, respectively. Additionally, the ARAUS dataset includes, for each augmented soundscape, pre-calculated acoustic and psychoacoustic features recommended by the ISO-12913. These features are calculated with ArtemiS SUITE [1], which is a proprietary software not easily available to researchers. As part of our work, we provide an open-source Python implementation of such features facilitating the reproducibility of the experiments[2].

3.2. Features

What follows is a description of the three aforementioned feature sets that we consider for our experiments.

Psychoacoustic features The standard ISO-12913 suggests a set of acoustic and psychoacoustic features to characterise urban soundscapes: sharpness, loudness, fluctuation strength, roughness, tonality, L_{Aeq} and L_{Ceq}. We compiled existing open source implementations for these features, and wrote custom implementations for the missing ones. For each feature, we use the statistics mean, maximum, and the 5th, 10th, 20th, 30th, 40th, 50th, 60th, 70th, 80th, 90th, and 95th percentiles calculated over time. Additionally, replicating ARAUS, the band powers summed over third-octave bands (5Hz to 20kHz) are included. This results in a total of 117 features. It should be noted that these features should not be computed directly on the WAV signal, but on the peak-Pascals pressure signal that results after applying a gain correction to the raw waveform. Thus, the waveform represents the SPL at which the signal was recorded, or, in this case, the level at which it was played in the listening tests. The L_{eq} value, provided in the ARAUS dataset, is used to calculate the mentioned calibration factor (designated as *wav gain*). This feature set is referred to as *ARAUS features*.

Signal processing features This set includes features typically used in traditional machine listening systems. *Freesound Extractor* algorithm from the Essentia audio analysis library[3] generates an extensive set of features from which we use: average loudness; loudness EBU-128; dynamic complexity; spectral flatness, roll-off, flux, skewness, spread, kurtosis and centroid; energy per bands (low, middle-low, middle-high, high); 13th first MFCCs; dissonance; zero-crossing rate; temporal centroid, kurtosis, skewness and spread; log attack-time; inharmonicity; and bpm. For each feature, we compute the statistics mean, variance, and the 20th and 80th percentiles over time, resulting in a total of 139 features. Contrary to the set above, these features are directly linked to the raw audio signal. However, a gain adjustment is performed to ensure that the signal

[1] https://www.head-acoustics.com/products

[2] https://github.com/MTG/soundlights

[3] https://essentia.upf.edu

amplitude proportions between different audio clips reflect the volume at which they were played during the listening tests. To achieve this, we apply the corresponding *wav gain* to each audio and then divide by a common normalization factor to prevent clipping. This set is referred to as *Freesound features*.

CLAP embeddings This set of features consists of the 512-length audio embedding generated using LAION-AI's CLAP model [24]. Since this model is trained using audio-text pairs, the resulting vector is expected to capture semantic information from the audio. This is unlike the *ARAUS* and *Freesound* feature sets, which only represent acoustic information from the sounds. The same scaling procedure used for the *Freesound* feature set is applied in this case. This representation is referred to as *CLAP features*.

The above feature sets are tested independently, but also in combination with information about which sound sources are present in the urban soundscape. Ideally, this information should include the predominant sound source. However, no dataset contains realistic urban soundscape sounds with both this source information and the pleasantness/eventfulness annotations. The ARAUS dataset provides the maskers (see Section 2) that were used to generate each augmented soundscape. Even though these sound sources might not always be predominant, it is guaranteed that they are present. Therefore, we use the maskers' information as a proxy for sound source information, and represent it with one-hot vectors. These six features are referred to as *sources features*.

3.3. Models

The emphasis of this work is not on the models to be trained but on the feature sets. Nevertheless, a number of preliminary experiments were carried out in which the performances of some classic machine-learning regression models were compared (like Support Vector Regression, Multi-layer Perceptron Regressor or regression based on K-Nearest-Neighbours). In these experiments, the best results were obtained by an Elastic Net model (as used in [18]), and a Random Forest Regressor. Therefore, these two models are implemented in our experiments using the Scikit-Learn library [4].

4. EVALUATION

To evaluate the predictive performance and robustness of the feature sets and models, we design a multi-faceted evaluation framework which not only includes the use of ARAUS data folds for cross-validation and model testing but it also involves the creation of a new testing set with data not present in the original dataset. Additionally, the analysis of models' robustness against sensor calibration is evaluated by introducing controlled variations in audio signals. Mean Absolute Error (MAE) is used as the main evaluation metric because it allows for a straightforward interpretation that represents the average absolute difference between the predicted and the ground-truth values.

4.1. Data folds

ARAUS includes five folds of augmented soundscapes for cross-validation and one test fold of 48 audios, reported under the labels *Val* and *fold-0* in Table 1, respectively. In addition, we create a complementary testing fold using 25 urban recordings downloaded from Freesound[5]. The selection was carried out manually by the authors and consists of 30-second excerpts of real urban environment recordings that include sources such as traffic, construction, rain, wind, voices, and music. Following ISO-12913, a listening test was carried out where 22 participants rated the 25 audios with 1-5 scales on how *pleasant*, *annoying*, *eventful*, *uneventful*, *vibrant*, *monotonous*, *chaotic* and *calm* the soundscapes were perceived. From those ratings, ground-truth pleasantness and eventfulness metrics were calculated following the same standard. The audios were calibrated and played at appropriate and varied L_{eq} values, regardless of the audio content. We refer to this fold as *fold-Fs*.

Feature set	Sound sources info	Model	Train	Val	Test fold-0	Test fold-Fs	Var. %
			PLEASANTNESS - MAE				
ARAUS	no	RFR	0.29	0.30	0.24	0.21	4.31
	yes	RFR	0.29	0.29	0.26	0.18	
Freesound	no	EN	0.29	0.30	0.22	0.19	2.19
	yes	EN	0.29	0.29	0.22	0.19	
CLAP	no	RFR	0.10	0.28	0.22	0.14	0.53
	yes	RFR	0.10	0.28	0.22	0.14	
			EVENTFULNESS - MAE				
ARAUS	no	EN	0.30	0.30	0.15	0.20	1.57
	yes	EN	0.30	0.30	0.14	0.20	
Freesound	no	RFR	0.13	0.29	0.16	0.22	0.02
	yes	RFR	0.13	0.29	0.16	0.22	
CLAP	no	RFR	0.10	0.29	0.20	0.18	-0.41
	yes	RFR	0.10	0.29	0.20	0.18	

Table 1: MAE results for the best performing models for the cross-validation folds and the two testing folds. The MAE variation percentage is included in the last column, representing the mean percentage variation in MAE when adding the sound sources information to the feature set (a positive percentage indicates an improvement in prediction). Note: *EN* and *RFR* stand for Elastic Net and Random Forest Regressor, respectively.

4.2. Robustness analysis

As has been mentioned, to evaluate the robustness of the studied models against different input signal calibration conditions, five controlled variations of the testing fold *fold-0* are generated by modifying the audio signals with *wav gain* adjustments of -6dB, +6dB, +12dB and +18dB; and a fifth variation with random *wav gain* within a fixed range [0-20dB].

5. RESULTS

Table 1 presents the MAE scores for the different combinations of models and feature sets evaluated. For pleasantness, the CLAP representation outperforms the other two feature sets in both test folds, reaching an MAE of 0.14 in *fold-Fs*. This indicates that, on a scale of [-1, 1], the predictions deviate by an average of 0.14. In terms of MAE variation resulting from the inclusion of the *source features*, the CLAP representation shows the smallest improvement,

[4] `https://scikit-learn.org`

[5] `https://freesound.org`

with just 0.53%, compared to 4.31% for *ARAUS features* and 2.19% for *Freesound features*. Regarding eventfulness, *ARAUS features* outperform the others when taking into account both test folds, but the smallest MAE for *fold-Fs*, 0.18 points, is achieved by *CLAP features*. When examining the MAE percentage variation, the inclusion of sound sources information has a smaller impact, with percentages closer to zero than those observed for pleasantness.

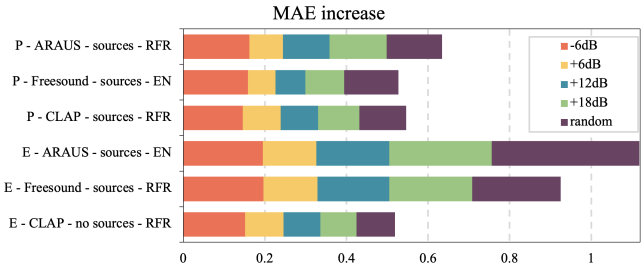


Figure 1: Increase in MAE value provoked by each *fold-0* variation with respect to the original and unvaried *fold-0* MAE.

Furthermore, all controlled calibration variations generated of *fold-0* result in higher MAE values. Figure 1 illustrates the increase of MAE only for the best-performing model for each feature set. The first noticeable observation is that the impact is greater on eventfulness, where the MAE increase is more pronounced. Also, it can be noted that the *ARAUS* feature set is more negatively affected in both cases, whereas the CLAP feature set appears to be the least affected. Among the variations, the 6dB increase in *wav gain* caused the smaller impact.

In terms of calculation time, *CLAP features* is the fastest set, taking 0.5s to calculate the embeddings for a 30s-long stereo audio file (sampled at 48kHz, run in a MacBook Pro M3). *Freesound features* and *ARAUS features* take 8x and 144x longer, respectively. Note that this comparison is limited as these feature sets are implemented in different frameworks and languages.

6. DISCUSSION

The experimental results indicate that, for predicting pleasantness, *CLAP features* outperform the other two sets, achieving an MAE of 0.22 and 0.14 for *fold-0* and *fold-Fs*, respectively. These results occur both when *CLAP features* are used alone and when combined with *sources features*, with only a 0.53% difference in performance between the two scenarios. Since CLAP embeddings intrinsically contain semantic information about the audio, additional sound source information is redundant. Conversely, for feature sets that lack this semantic data, including the source information positively impacts the accuracy in the prediction of pleasantness: the performances of *ARAUS* and *Freesound* feature sets improve by 4.31% and 2.19%, respectively. These findings suggest a clear correlation between the sound sources that are present in the urban soundscape and the perceived sensation of pleasantness. In fact, these results coincide with those obtained in the listening test. A quantitative analysis, which can be seen in Figure 2, shows a clear source-class separation on the pleasantness scale depending on the predominant sound source: construction and traffic noises are positioned on the negative side of the axis, while natural sounds are on the positive side.

For predicting eventfulness, all feature sets perform similarly, with *ARAUS features* showing slightly better results when considering the MAE mean of both test sets. Besides, the impact of the inclusion of *source features* is negligible, being smaller than 2% for *ARAUS features*, and close to zero for *Freesound* and *CLAP* feature sets. This indicates a weaker correlation between the sound sources present in the soundscape and the sensation of eventfulness, coinciding again with the data extracted from the listening test, where there is more overlap between class groups when seen from the eventfulness axis (see Figure 2).

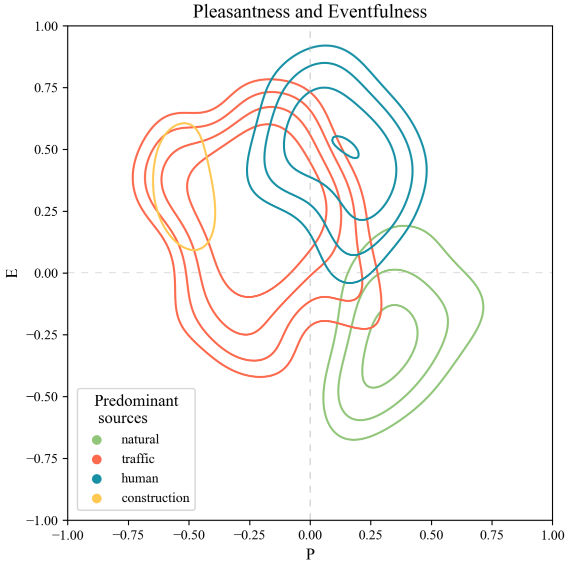


Figure 2: Two-dimensional Kernel Density Estimate plot of the pleasantness(P) and eventfulness(E) values reported from the answers to the listening test.

In terms of robustness against changes in sensor calibration, none of the trained models demonstrate strong capabilities, as MAE increases notably in every *fold-0* variation case. Nevertheless, predictions of eventfulness are more negatively affected, potentially indicating a correlation between SPL, or loudness, and the perception of eventfulness. Moreover, models trained with *CLAP features* seem to be slightly less impacted by the calibration changes. In addition to this, their rapid generation time suggests that *CLAP features* are adequate for real-time contexts.

7. CONCLUSION

This research shows that CLAP embeddings generated by LAION-AI's CLAP model demonstrate high performance as input to models for predicting *pleasantness* and *eventfulness* perceptual sound qualities. Even though the sound representation does not present strong robustness to variations in sensor calibration, it can be computed rapidly, making it suitable for real-time applications. Moreover, our study indicates a clear correlation between the sound sources present in an urban soundscape and its sensation of pleasantness. Future research directions could include evaluating the developed models in the context of a real-world acoustic sensor network and incorporating sound classification and source separation technologies to improve the models' accuracy and capabilities for meaningful soundscape characterisation and monitoring.

8. ACKNOWLEDGMENTS

This work was supported by the project "Soundlights: Distributed Open Sensors Network and Citizen Science for the Collective Management of the City's Sound Environments" (9382417), funded by *BIT Habitat (Ajuntament de Barcelona)* under the program *La Ciutat Proactiva*; and by the *IA y Música: Cátedra en Inteligencia Artificial y Música* (TSI-100929-2023-1) by the *Secretaría de Estado de Digitalización e Inteligencia Artificial* and *NextGenerationEU* under the program *Cátedras ENIA 2022*. Additionally, we would like to acknowledge *Bitlab Cooperativa Cultural* for the fruitful discussions in the context of the Soundlights project.

9. REFERENCES

[1] European Parliament and Council, "Directive 2002/49/EC of the European Parliament and of the Council of 25 June 2002 relating to the assessment and management of environmental noise - Declaration by the Commission in the Conciliation Committee on the Directive relating to the assessment and management of environmental noise," 2002, Official Journal of the European Union, L 189, p. 12–25. [Online]. Available: https://eur-lex.europa.eu/eli/dir/2002/49/oj

[2] M. Raimbault and D. Dubois, "Urban soundscapes: Experiences and knowledge," *Cities*, vol. 22, no. 5, pp. 339–350, 2005.

[3] J. M. Morillas, V. G. Escobar, G. R. Gozalo, R. Vílchez-Gómez, J. A. M. Sierra, J. T. Carmona, C. P. Gajardo, and F. J. C. D. Río, "Sound quality in urban environments and its relationship with acoustic parameters," in *Noise Control and Acoustics Division Conference (NCAD)*, 2013.

[4] B. Truax, "Environmental sound and its relation to human emotion," *Canadian Acoustics*, vol. 44, no. 3, 2016.

[5] A. Fiebig, P. Jordan, and C. C. Moshona, "Assessments of acoustic environments by emotions – the application of emotion theory in soundscape," *Frontiers in Psychology*, vol. 11, 2020.

[6] International Organization for Standardization, "ISO 12913-1. Acoustics-Soundscape-Part 1: Definition and conceptual framework," 2014. [Online]. Available: www.iso.org

[7] ——, "ISO 12913-2. Acoustics-Soundscape-Part 2: Data collection and reporting requirements," 2018. [Online]. Available: www.iso.org

[8] ——, "ISO 12913-3. Acoustics-Soundscape-Part 3: Data analysis," 2019. [Online]. Available: www.iso.org

[9] A. Mehrabian and J. A. Russell, *An approach to environmental psychology.* The MIT Press, 1974.

[10] J. A. Russell, "A circumplex model of affect," *Journal of Personality and Social Psychology*, vol. 39, no. 6, pp. 1161–1178, 1980.

[11] Österten Axelsson, M. E. Nilsson, and B. Berglund, "A principal components model of soundscape perception," *The Journal of the Acoustical Society of America*, vol. 128, no. 5, pp. 2836–2846, 2010.

[12] J. Fan, F. Tung, W. Li, and P. Pasquier, "Soundscape emotion recognition via deep learning," in *Sound and Music Computing Conference (SMC)*, 2018.

[13] J. Fan, M. Thorogood, and P. Pasquier, "Emo-soundscapes: A dataset for soundscape emotion recognition," in *International Conference on Affective Computing and Intelligent Interaction (ACII)*, 2017.

[14] "Emo-Soundscapes — Dataset." [Online]. Available: https://www.metacreation.net/projects/emo-soundscapes/

[15] "ATHUS (Athens Urban Soundscape) - Dataset." [Online]. Available: https://users.iit.demokritos.gr/~tyianak/soundscape/

[16] T. Giannakopoulos, M. Orfanidi, and S. Perantonis, "Athens Urban Soundscape (ATHUS): A Dataset for Urban Soundscape Quality Recognition," in *Multimedia Modelling (MMM)*, 2019.

[17] "ARAUS (Affective Responses to Augmented Urban Soundscapes) - Dataset." [Online]. Available: https://researchdata.ntu.edu.sg/dataset.xhtml?persistentId=doi:10.21979/N9/9OTEVX

[18] K. Ooi, Z. T. Ong, K. N. Watcharasupat, B. Lam, J. Y. Hong, and W. S. Gan, "ARAUS: A Large-Scale Dataset and Baseline Models of Affective Responses to Augmented Urban Soundscapes," *IEEE Transactions on Affective Computing*, vol. 15, pp. 105–120, 2024.

[19] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *ACM Multimedia Conference (MM)*, 2014.

[20] "UrbanSound - Dataset." [Online]. Available: https://urbansounddataset.weebly.com/urbansound.html

[21] "UrbanSound8K - Dataset." [Online]. Available: https://urbansounddataset.weebly.com/urbansound8k.html

[22] J. Salamon, D. MacConnell, M. Cartwright, P. Li, and J. P. Bello, "Scaper: A library for soundscape synthesis and augmentation," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2017.

[23] R. O. Araz, D. Bogdanov, P. Alonso-Jiménez, and F. Font, "Evaluation of deep audio representations for semantic sound similarity," in *International Conference on Content-based Multimedia Indexing (CBMI)*, In Press.

[24] "LAION-AI/CLAP Github Repository." [Online]. Available: https://github.com/LAION-AI/CLAP

[25] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, "Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.

[26] "microsoft/CLAP Github Repository." [Online]. Available: https://github.com/microsoft/CLAP

[27] B. Elizalde, S. Deshmukh, M. A. Ismail, and H. Wang, "Clap learning audio concepts from natural language supervision," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.

DATA-EFFICIENT LOW-COMPLEXITY ACOUSTIC SCENE CLASSIFICATION IN THE DCASE 2024 CHALLENGE

Florian Schmid[1], *Paul Primus*[1], *Toni Heittola*[3],
Annamaria Mesaros[3], *Irene Martín-Morató*[3], *Khaled Koutini*[2], *Gerhard Widmer*[1,2]

[1]Institute of Computational Perception, Johannes Kepler University Linz, Austria
[2]LIT Artificial Intelligence Lab, Linz, Austria, [3]Computing Sciences, Tampere University, Finland
{florian.schmid, paul.primus, gerhard.widmer}@jku.at
{toni.heittola, annamaria.mesaros, irene.martinmorato}@tuni.fi

ABSTRACT

This article describes the *Data-Efficient Low-Complexity Acoustic Scene Classification* Task in the DCASE 2024 Challenge and the corresponding baseline system. The task setup is a continuation of previous editions (2022 and 2023), which focused on recording device mismatches and low-complexity constraints. This year's edition introduces an additional real-world problem: participants must develop data-efficient systems for five scenarios, which progressively limit the available training data. The provided baseline system is based on an efficient, factorized CNN architecture constructed from inverted residual blocks and uses Freq-MixStyle to tackle the device mismatch problem. The task received 37 submissions from 17 teams, with the large majority of systems outperforming the baseline. The top-ranked system's accuracy ranges from 54.3% on the smallest to 61.8% on the largest subset, corresponding to relative improvements of approximately 23% and 9% over the baseline system on the evaluation set.

Index Terms— DCASE Challenge, Acoustic Scene Classification, data-efficiency, low-complexity, multiple devices

1. INTRODUCTION

Acoustic Scene Classification (ASC) aims at detecting the environmental context in which audio was captured, based on a short excerpt [1]. The environmental context is given as a set of pre-defined acoustic scene classes such as *Metro station*, *Urban park*, or *Public square*. Since its inception, the ASC task has been an integral part of the DCASE Challenge. Each year's edition focused on one or multiple challenging machine-learning aspects in addition to the supervised classification task itself. These aspects include open-set classification [2], constraints on the model's size and computational complexity [3–5], and generalization across different recording devices [3,6]. These additional problems target the real-world applicability of ASC systems; for instance, the methods should be robust to diverse recording devices and sufficiently lightweight to be deployable on embedded devices. In the 2024 edition[1] of the ASC task, an additional challenging real-world aspect is addressed: the limited availability of training data. This setting intends to spark research on data-efficient learning methods capable of achieving high classification performance given only a small number of labeled acoustic scene examples for training.

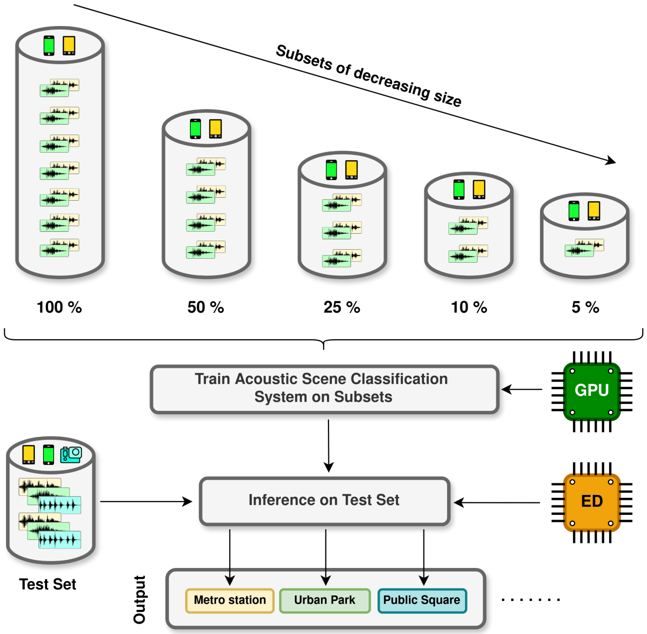


Figure 1: Overview of Data-Efficient Low-Complexity Acoustic Scene Classification. Submitted systems must be trained on five datasets of varying sizes, they must generalize to unseen recording devices, and they are required to be lightweight enough for inference on an embedded device (ED).

Figure 1 shows an overview of the task setup. The ASC systems must be trained on subsets of a fixed training set that progressively limit the number of training samples, where the smallest subset only contains 5% of the audio snippets in the full training set (see Section 3.2). The training procedure is not limited in terms of complexity and may be executed on high-end GPU hardware. However, aligned with real-world requirements, the system must be lightweight for inference such that it can be deployed on embedded devices (see Section 3.3). Additionally, the developed ASC system must be robust to unseen recording devices. To test this ability, the test set includes audio clips recorded by new devices that are not available in the training sets (see Section 3.2).

The rest of this paper is structured as follows: Section 2 briefly discusses the role of low-complexity constraints and the device generalization problem in previous editions of the task. Section 3 intro-

[1] Task Description Page: `https://dcase.community/challenge2024/task-data-efficient-low-complexity-acoustic-scene-classification`

duces the setup for Data-Efficient Low-Complexity Acoustic Scene Classification in the DCASE 2024 Challenge; the baseline system is presented in Section 4. The outcome of the challenge is analyzed in Section 5 and the paper is concludes in Section 6.

2. PREVIOUS EDITIONS

The low-complexity aspect has already been investigated in previous DCASE challenges and has undergone several refinements. In the 2020 [3] and 2021 [4] editions, systems were limited with respect to model size, allowing 500 kB and 128 kB for non-zero parameters, respectively. In the 2022 edition [5], the complexity constraint additionally included computational complexity, allowing a maximum of 30 MMACs (million multiply-accumulate operations), modeled after Cortex-M4 devices. The maximum number of parameters was 128K, with the variable type fixed to INT8. The 2023 edition took this one step further and included model size and computational complexity as part of the ranking metric, requiring participants to tune the system's performance–complexity trade-off. In response to the low-complexity requirements, training techniques such as Sparsification [7], Pruning [8], Quantization [9], or Knowledge Distillation [10] have been extensively studied, and efficient factorized CNN architectures [11–13] have been designed.

Besides low-complexity techniques, substantial research has been conducted on the device mismatch problem. Efforts to improve device generalization involved suppressing device information via normalization [9] and domain adaptation [14], balancing the devices by changing the sampling distribution [15] and augmenting audio segments with device translators [9], Freq-MixStyle [10, 16], and device impulse response augmentation [17].

3. TASK SETUP

While low complexity and generalization across different recording devices are well-studied topics, the specific aspect of interest in the 2024 edition is the limited availability of acoustic scene data for training. Specifically, participants were encouraged to develop data-efficient systems and study techniques that can alleviate the data scarcity problem, such as using extensive audio augmentation methods, transferring knowledge from general-purpose audio datasets, or incorporating well-suited inductive biases.

3.1. Dataset

The task builds on top of the *TAU Urban Acoustic Scenes 2022 Mobile* dataset [3, 6], which was used in the 2022 and 2023 editions of the task [5]. The dataset provides one-second audio snippets with a sampling rate of 44.1 kHz in single-channel, 24-bit format and consists of recordings from ten distinct acoustic scenes.

The audio was recorded in multiple European cities with four recording devices in parallel. The primary device, referred to as device *A*, is a high-quality binaural device, while *B*, *C* and *D* are commonly available consumer devices. Additionally, 10 simulated devices (*S1-S10*) are created using audio from device A and a set of impulse responses from mobile devices. For details on the dataset creation and the exact distribution of devices, please refer to [3].

The data is split into a development and an evaluation set. The development set, consisting of 64 hours of audio, contains 3 real devices (A, B, C) and 6 simulated devices (S1–S6). The evaluation set comprises five unseen devices (D and S7-S10) and two unseen cities, in addition to devices and cities overlapping with the development set. The evaluation set is used to rank submissions and therefore comes without corresponding scene labels. Device and city information is not provided for recordings in the evaluation set.

3.2. Data-Efficient Evaluation

The development set used for the 2024 challenge is the same one as used in the previous two years and described above. It comes with the same pre-defined split into a development-train and a development-test partition. The development-train set contains six devices (A, B, C, S1-S3), leaving three unseen devices (S4-S6) for the development-test set to measure the device generalization performance.

For the evaluation of data efficiency, this year's setup introduces five pre-defined subsets that progressively limit the available training data and contain 100%, 50%, 25%, 10%, and 5% of the recordings in the development-train set. The distribution of acoustic scenes, cities, and recording devices is kept similar across all subsets. The smaller subsets are fully included in the larger ones, corresponding to the idea of progressively collecting more data.

Participants are allowed to submit up to three different systems that may be specialized for the different training set sizes. Each system must be trained on all five subsets, and the performances on the development-test set must be reported. A system is considered to be the same if its architecture and design choices (such as building blocks, features, data augmentation techniques, decision-making, etc.) remain the same. However, basic hyperparameters like the number of update steps, learning rate, batch size, or regularization strength may vary for training on the different subsets.

All systems must be trained only on the respective subset and the explicitly allowed external resources. The allowed external resources include general-purpose audio datasets, such as AudioSet [18] or FSD50K [19], but no datasets specific to acoustic scenes.

The leaderboard ranking score is computed as follows. First, class-wise macro-averaged accuracies for all $P = 5$ development-train subsets and all N submissions are computed. The accuracy of the n-th submission on the $p\%$ subset is denoted as $ACC_{n,p}$. The scores are then aggregated by choosing the best-performing system for each subset and averaging the resulting accuracies.

$$\text{score} := \frac{1}{P} \sum_{p \in \{5,10,25,50,100\}} \max_{n \in \{1,\ldots,N\}} ACC_{n,p} \tag{1}$$

The outlined setup encourages research into the following scientific questions: how does the performance of systems vary with the number of available labeled training samples? how can systems be adapted to better cope with the limited availability of labeled training data? how can general-purpose audio datasets be exploited to mitigate the need for larger amounts of acoustic scenes?

3.3. System Complexity Requirements

The system complexity is limited in terms of model size and MMACs. The maximum memory allowance for model parameters is 128 kB, with no requirement regarding the numerical representation. That is, participants can trade off the number of parameters and the numerical representation. For example, the memory limit translates to 128K parameters when using 8-bit quantization, or 32K parameters when using 32-bit precision. The computational complexity is limited to 30 MMACs for the inference on a one-second audio segment. These complexity limits are modeled after Cortex-M4 devices (e.g., STM32L496@80MHz or Arduino Nano 33@64MHz).

System Label	Score	Team Rank	Size	MACs	Architecture	Complexity	Dev. Gen.	External
Han_SJTUTHU_task1_2	58.2	**1**	126kB	29M	SSCP-Mobile	fp16, KD, prun.	FMS	PaSST
Shao_NEPUMSE_task1_1	57.2	**3**	107kB	16M	IRMamba	int8, KD	FMS, DIR	PaSST
MALACH24_JKU_task1_1	57.0	**2**	122kB	29M	CP-Mobile	fp16, KD	FMS, DIR	AudioSet
Yeo_NTU_task1_2	56.1	**5**	122kB	29M	CP-Mobile	fp16, KD	FMS, DIR	PaSST
Cai_XJTLU_task1_3	56.0	**4**	126kB	29M	TF-SepNet	int8, KD	FMS, DIR	AudioSet
Park_KT_task1_2	55.4	**6**	126kB	26M	GhostRes2Net	fp16, KD	FMS, DIR	PaSST, EAT
OO_NTUPRDCSG_task1_1	54.8	**7**	116kB	29M	MofleNet	int8	FMS, DIR	-
Werning_UPBNT_task1_1	54.4	**8**	122kB	29M	CP-Mobile	fp16, KD	FMS	AudioSet
Truchan_LUH_task1_1	53.1	**9**	94kB	29M	Isotropic	fp16	FMS, DIR	-
Yan_NPU_task1_1	52.9	**10**	124kB	29M	MAR-CNN	fp32	FMS	-
Baseline	50.7		122kB	29M	CP-Mobile	fp16	FMS	-

Table 1: This table lists the top-ten teams' best systems according to their evaluation set performance. **Team Rank** indicates the team's overall rank, which is based on multiple submitted systems, and **Score** is the average accuracy across all splits of the respective system listed in the table. int8, fp16, and fp32 refer to the numerical precision of model parameters for inference, corresponding to 8, 16, and 32 bits, respectively. KD, FMS, and DIR are abbreviations for Knowledge Distillation, Freq.-MixStyle, and Device Impulse Response augmentation, respectively, and the column **External** indicates external resources used.

4. BASELINE SYSTEM

The baseline system is a simplified version of the top-ranked system submitted to the 2023 edition [20]. It is based on a receptive-field-regularized, factorized CNN design. Audio input is resampled to 32 kHz and converted to mel spectrograms using a 4096-point FFT with a window size of 96 ms and a hop size of approximately 16 ms, followed by a mel transformation with a filterbank of 256 mel bins. The system is trained for 150 epochs using the AdamW optimizer and a batch size of 256. Freq-MixStyle [10, 16] is applied to tackle the device mismatch problem, and time rolling of the waveform and frequency masking are used to augment the training data. The baseline system requires 29.4 MMACs for the inference on a one-second audio clip. The memory required for the model parameters amounts to 122.3 kB, resulting from the 61,148 parameters used in 16-bit precision (fp16).

The baseline's accuracy on the development-test split ranges from 42.40% for the smallest training subset (5%) to 56.99% accuracy for the full set (100%). The performance increases monotonically as the number of audio segments available for training increases. The code and a detailed description of the baseline system are available online[2].

5. CHALLENGE RESULTS

The task received 37 submissions from 17 teams and is therefore the second most popular task in the 2024 edition of the DCASE challenge. The slight decrease in popularity compared to the previous year's edition is likely due to the more complex setup. 16 out of 17 teams outperformed the baseline system and for most of the teams, the performance on the development-test split aligns well with the performance on the evaluation set. The challenge website contains detailed results and descriptions on all submitted systems[3].

Table 1 presents the best systems submitted by the ten top-ranked teams and lists details in terms of architectures, complexity handling, device generalization, and usage of external resources. *Score* denotes the average accuracy across all five training set splits on the evaluation set. Note that a team's rank depends on all three allowed submissions, rather than only on the system achieving the highest score (which is why the *Team Rank* column of Table 1 is not perfectly sorted).

5.1. Architectures

In response to the low-complexity constraints and following the trend observed in the previous edition of this task [5], the large majority of systems are based on factorized CNN architectures. Most prominently, factorization is realized via inverted residual blocks, as used in the baseline architecture. Table 1 shows that four out of the ten best systems are based on modified versions of the CP-Mobile architecture [20]. The top-ranked system [21] further reduces CP-Mobile's complexity by factorizing the spatial convolutions with a 3x3 kernel into two separate convolutions with 1x3 and 3x1 kernels. Team *Shao_NEPUMSE* [22] enhances an inverted residual block-based architecture with parallel Mamba blocks [23], a derivative of state space models. Teams *Cai_XJTLU* [24] and *Park_KT* [25] use modified versions TF-SepNet [26] and BCRes2Net [9], respectively, both of which achieved high ranks in previous editions of this task and decouple spatial convolutions over frequency and time dimensions. Team *OO_NTUPRDCSG* [27] introduces MofleNet by enhancing the CP-Mobile architecture with channel shuffle operations; Team *Truchan_LUH* [28] uses an isotropic convolutional architecture following a patch embedding layer; and Team *Yan_NPU* [29] presents MAR-CNN, an asymmetric multi-branch convolutional architecture.

5.2. System Complexity

Knowledge Distillation (KD) can be identified as the most prominent technique to tackle the low-complexity constraints, with the six top-ranked teams using KD. The most popular teacher model is the audio spectrogram transformer PaSST [30]. Among other models that proved to be successful teachers are CP-ResNet [31] (Teams *MALACH24_JKU* [32] and *Shao_NEPUMSE* [22]), BEATs [33] (Team *Cai_XJTLU* [24]), EAT [34] (Team *Park_KT* [25]) and DyMN [35] (Team *Bai_JLESS* [36]). Regarding numerical representation of parameters, both 8-bit and 16-bit precision solutions are among the top-ranked systems. To convert parameters to 8-bit pre-

[2] Source Code: `https://github.com/CPJKU/dcase2024_task1_baseline/tree/main`

[3] Results: `https://dcase.community/challenge2024/task-data-efficient-low-complexity-acoustic-scene-classification-results`

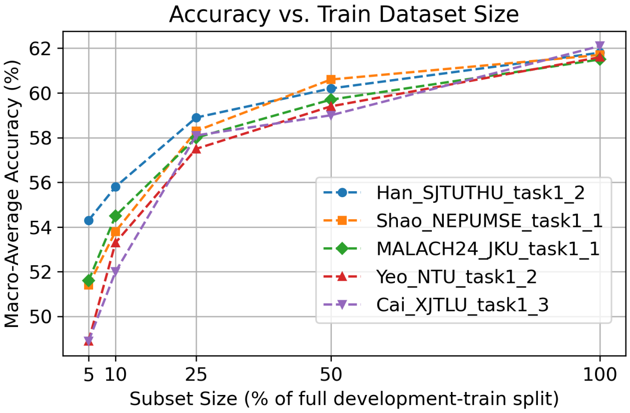


Figure 2: Performance of the best systems from the five top-ranked teams on the evaluation set for training on the five subsets (5%, 10%, 25%, 50%, 100%) of the development-train split.

cision, Teams *Shao_NEPUMSE* [22] and *OO_NTUPRDCSG* [27] use Quantization-Aware-Training, while Team *Cai_XJTLU* [24] shows that also Post-Training Static Quantization can lead to good results. In addition to KD, the top-ranked system by Team *Han_SJTUTHU* [21] uses pruning. They construct a large version of SSCP-Mobile by increasing the number of channels, apply pruning to meet complexity constraints, and then fine-tune the pruned model using KD.

5.3. Device Generalization

The majority of teams tackle the device mismatch with dedicated data augmentation techniques. In this regard, Freq-MixStyle [10, 16], which is also integrated into the baseline system, is used by all of the ten top-ranked teams. Additionally, seven out of the ten top-ranked systems use device impulse response augmentation, implemented using convolution with 66 freely available impulse responses from MicIRP[4]. An interesting alternative is presented by Team *Truchan_LUH* [28] using an adversarial device classifier that forces the feature extractor to learn device-invariant representations.

5.4. Limited Training Data

Figure 2 shows that the top systems submitted to the challenge perform similarly when trained on the 100% train split. However, the smaller the size of the training set, the larger the performance differences, underscoring the large impact of creating data-efficient systems. In fact, the top-ranked system does not achieve the highest accuracy for the 50% and 100% training splits, but it surpasses other systems on the 5%, 10%, and 25% subsets. In particular, on the smallest training set, it outperforms all other teams' systems by more than 2 percentage points in terms of accuracy.

In the following, we describe approaches by participants to counteract the performance dropoff for small training sets.

General-Purpose Audio Datasets: Very commonly, participants make use of large general-purpose audio datasets, in particular, AudioSet [18], to alleviate the data scarcity problem. This is achieved in three different ways: (1) by fine-tuning a large, pre-trained model on ASC and using it as a teacher model in a KD setup; (2) by directly pre-training a low-complexity model on AudioSet; and (3) by extracting audio clips from AudioSet as additional training data. The effectiveness of (1) is underlined by the fact that most of the top-ranked teams use an AudioSet pre-trained transformer model as a teacher in a KD setup. For example, Team *Cai_XJTLU* [24] achieves an accuracy of 55.7% on the development-test set when fine-tuning multiple BEATs [33] models on the 5% training subset, which is higher than the Baseline system's accuracy using 10 times as much training data. Regarding (2), the team with the second-best performance on the 5% and 10% subsets, Team *MALACH24_JKU* [32], pre-trains CP-Mobile on AudioSet and reports a large performance gain for fine-tuning on smaller training subsets. Concerning (3), Team *Werning_UPBNT* [37] trains a dataset domain classifier to extract audio clips from AudioSet that are similar to the samples in the respective training sets and uses these as additional samples for KD. Additionally, Team *Surkov_IMTO* [38] selects AudioSet clips from specific event classes such as *Bus* or *Train* and uses them as additional unlabeled samples in a mean-teacher approach.

Extensive Data Augmentation: Besides Freq-MixStyle and DIR augmentation, extensive data augmentation is applied to improve generalization performance on the small training sets. In this regard, Team *MALACH24_JKU* [32] uses FilterAugment [39], Team *Shao_NEPUMSE* [22]) experiments with audio playback, Team *Chen_SCUT* [40] uses Spectrum Modulation. SpecAugment, time rolling, and Mixup are widely used throughout submissions.

Model Size and Architecture: Team *Yeo_NTU* [41] investigated the relationship between model size and performance on small training splits and found that models of reduced complexity generalize better for small training splits. Team *Park_KT* [25] enhanced their network with Snake activation functions and showed that the introduced inductive bias on periodicity leads to a large performance gain on smaller training sets.

6. CONCLUSION

This paper has presented an analysis of Task 1 in the DCASE 2024 challenge, which focused on the real-world deployment of ASC systems with low-complexity constraints, device mismatch, and training data scarcity being the main hurdles to overcome. The task remained the second most popular in the DCASE 2024 challenge, underscoring the high interest in the task despite the increasingly challenging setup. Multiple strategies have been proposed to tackle the limited availability of training data; most highly-performing systems transferred knowledge from a large general-purpose audio dataset to the ASC task, either in the form of pre-trained models or by extracting additional ASC-related audio clips for training. Data augmentation remained a highly important aspect, not only to address device generalization but also to improve generalization capabilities, with only a small training set available. Other solutions to the data scarcity problem involve adapting the model's complexity or building inductive biases into the model architecture. Summarizing the output of the task, several promising techniques have been proposed that can boost performance on downstream tasks when only a small training set is available.

7. ACKNOWLEDGMENT

The LIT AI Lab is supported by the Federal State of Upper Austria. Gerhard Widmer's work is supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme, grant agreement No 101019375 (Whither Music?).

[4] http://micirp.blogspot.com/

8. REFERENCES

[1] E. Benetos, D. Stowell, and M. D. Plumbley, "Approaches to complex sound scene analysis," in *Cham: Springer International Publishing*, 2018.

[2] A. Mesaros, T. Heittola, and T. Virtanen, "Acoustic scene classification in DCASE 2019 challenge: Closed and open set classification and data mismatch setups," in *DCASE Workshop*, 2019.

[3] T. Heittola, A. Mesaros, and T. Virtanen, "Acoustic scene classification in DCASE 2020 challenge: Generalization across devices and low complexity solutions," in *DCASE Workshop*, 2020.

[4] I. Martín-Morató, T. Heittola, A. Mesaros, and T. Virtanen, "Low-complexity acoustic scene classification for multi-device audio: Analysis of DCASE 2021 challenge systems," in *DCASE Workshop*, 2021.

[5] I. Martín-Morató, F. Paissan, A. Ancilotto, T. Heittola, A. Mesaros, E. Farella, A. Brutti, and T. Virtanen, "Low-complexity acoustic scene classification in DCASE 2022 challenge," in *DCASE Workshop*, 2022.

[6] A. Mesaros, T. Heittola, and T. Virtanen, "A multi-device dataset for urban acoustic scene classification," in *DCASE Workshop*, 2018.

[7] C.-H. H. Yang, H. Hu, S. M. Siniscalchi, Q. Wang, W. Yuyang, X. Xia, Y. Zhao, Y. Wu, Y. Wang, J. Du, and C.-H. Lee, "A lottery ticket hypothesis framework for low-complexity device-robust neural acoustic scene classification," DCASE Challenge, Tech. Rep., 2021.

[8] K. Koutini, J. Schlüter, and G. Widmer, "CPJKU submission to DCASE21: Cross-device audio scene classification with wide sparse frequency-damped CNNs," DCASE Challenge, Tech. Rep., 2021.

[9] B. Kim, S. Yang, J. Kim, and S. Chang, "QTI submission to DCASE 2021: Residual normalization for device-imbalanced acoustic scene classification with efficient design," DCASE Challenge, Tech. Rep., 2021.

[10] F. Schmid, S. Masoudian, K. Koutini, and G. Widmer, "CP-JKU submission to DCASE22: Distilling knowledge for low-complexity convolutional neural networks from a patchout audio transformer," DCASE Challenge, Tech. Rep., 2022.

[11] J. Tan and Y. Li, "Low-complexity acoustic scene classification using blueprint separable convolution and knowledge distillation," DCASE Challenge, Tech. Rep., 2023.

[12] Y. Cai, M. Lin, C. Zhu, S. Li, and X. Shao, "DCASE2023 task1 submission: Device simulation and time-frequency separable convolution for acoustic scene classification," DCASE Challenge, Tech. Rep., 2023.

[13] F. Schmid, T. Morocutti, S. Masoudian, K. Koutini, and G. Widmer, "CP-JKU submission to DCASE23: Efficient acoustic scene classification with cp-mobile," DCASE Challenge, Tech. Rep., 2023.

[14] K. Koutini, F. Henkel, H. Eghbal-zadeh, and G. Widmer, "CP-JKU submissions to DCASE'20: Low-complexity cross-device acoustic scene classification with RF-regularized CNNs," DCASE Challenge, Tech. Rep., 2020.

[15] J.-H. Lee, J.-H. Choi, P. M. Byun, and J.-H. Chang, "Hyu submission for the DCASE 2022: Efficient fine-tuning method using device-aware data-random-drop for device-imbalanced acoustic scene classification," DCASE Challenge, Tech. Rep., 2022.

[16] B. Kim, S. Yang, J. Kim, H. Park, J. Lee, and S. Chang, "Domain generalization with relaxed instance frequency-wise normalization for multi-device acoustic scene classification," in *Interspeech*, 2022.

[17] T. Morocutti, F. Schmid, K. Koutini, and G. Widmer, "Device-robust acoustic scene classification via impulse response augmentation," in *EUSIPCO*, 2023.

[18] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *ICASSP*, 2017.

[19] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "FSD50K: an open dataset of human-labeled sound events," *IEEE ACM Trans. Audio Speech Lang. Process.*, 2022.

[20] F. Schmid, T. Morocutti, S. Masoudian, K. Koutini, and G. Widmer, "Distilling the knowledge of transformers and CNNs with CP-mobile," in *DCASE Workshop*, 2023.

[21] H. Bing, H. Wen, C. Zhengyang, J. Anbai, C. Xie, F. Pingyi, L. Cheng, L. Zhiqiang, L. Jia, Z. Wei-Qiang, and Q. Yanmin, "Data-efficient acoustic scene classification via ensemble teachers distillation and pruning," DCASE Challenge, Tech. Rep., 2024.

[22] Y.-F. Shao, P. Jiang, and W. Li, "Low-complexity acoustic scene classification with limited training data," DCASE Challenge, Tech. Rep., 2024.

[23] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," *CoRR*, vol. abs/2312.00752, 2023.

[24] Y. Cai, M. Lin, S. Li, and X. Shao, "DCASE2024 task1 submission: Data-efficient acoustic scene classification with self-supervised teachers," DCASE Challenge, Tech. Rep., 2024.

[25] J. Park, T. Kim, D. Rho, J. Kim, and G. Lee, "Kt submission: Periodic activation and knowledge distillation for data-efficient low-complexity acoustic scene classification," DCASE Challenge, Tech. Rep., 2024.

[26] Y. Cai, P. Zhang, and S. Li, "TF-SepNet: An efficient 1d kernel design in CNNs for low-complexity acoustic scene classification," in *ICASSP*, 2024.

[27] Y. Oo and N. Srikanth, "Low complexity acoustic scene classification with moflenet," DCASE Challenge, Tech. Rep., 2024.

[28] H. Truchan, T. H. Ngo, and Z. Ahmadi, "Ascdomain: Domain invariant device-adversarial isotropic convolutional neural architecture," DCASE Challenge, Tech. Rep., 2024.

[29] C. Yan, Y. Yu, and X. Xiong, "Submission for DCASE 2024 task1: An asymmetric residual deep neural network for low-complexity acoustic scene classification," DCASE Challenge, Tech. Rep., 2024.

[30] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, "Efficient training of audio transformers with patchout," in *Interspeech*, 2022.

[31] K. Koutini, H. Eghbal-zadeh, and G. Widmer, "Receptive field regularization techniques for audio classification and tagging with deep convolutional neural networks," *IEEE ACM Trans. Audio Speech Lang. Process.*, 2021.

[32] N. David, R. Aida, and S. Patrick, "Data-efficient acoustic scene classification with pre-trained CP-Mobile," DCASE Challenge, Tech. Rep., 2024.

[33] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, Z. Chen, W. Che, X. Yu, and F. Wei, "BEATs: Audio pre-training with acoustic tokenizers," in *ICML*, 2023.

[34] W. Chen, Y. Liang, Z. Ma, Z. Zheng, and X. Chen, "EAT: self-supervised pre-training with efficient audio transformer," *CoRR*, 2024.

[35] F. Schmid, K. Koutini, and G. Widmer, "Dynamic convolutional neural networks as efficient pre-trained audio models," *IEEE ACM Trans. Audio Speech Lang. Process.*, 2024.

[36] J. Bai, M. Wang, E.-L. Tan, J. J. S. Yeo, J. W. Yeow, S. Peksi, D. Shi, W.-S. Gan, and J. Chen, "Hierarchical acoustic scene classification with knowledge distillation and pre-trained dynamic networks," DCASE Challenge, Tech. Rep., 2024.

[37] A. Werning and R. Haeb-Umbach, "Upb-nt submission to DCASE24: Dataset pruning for targeted knowledge distillation," DCASE Challenge, Tech. Rep., 2024.

[38] M. Surkov, "Efficient acoustic scene classification using mean-teacher and knowledge distillation," DCASE Challenge, Tech. Rep., 2024.

[39] H. Nam, S. Kim, and Y. Park, "Filteraugment: An acoustic environmental data augmentation method," in *ICASSP*, 2022.

[40] G. Chen and Y. Li, "Data-efficient low-complexity acoustic scene classification using parallel attention broad-cast-residual network," DCASE Challenge, Tech. Rep., 2024.

[41] S. Yeo, E.-L. Tan, J. Bai, S. Peksi, and W.-S. Gan, "Data efficient acoustic scene classification using sing teacher-informed confusing class instruction," DCASE Challenge, Tech. Rep., 2024.

MULTI-ITERATION MULTI-STAGE FINE-TUNING OF TRANSFORMERS FOR SOUND EVENT DETECTION WITH HETEROGENEOUS DATASETS

Florian Schmid[1], *Paul Primus*[1], *Tobias Morocutti*[1], *Jonathan Greif*[1], *Gerhard Widmer*[1,2]

[1]Institute of Computational Perception (CP-JKU), [2]LIT Artificial Intelligence Lab,
Johannes Kepler University Linz, Austria
{florian.schmid, paul.primus}@jku.at

ABSTRACT

A central problem in building effective sound event detection systems is the lack of high-quality, strongly annotated sound event datasets. For this reason, Task 4 of the DCASE 2024 challenge proposes learning from two heterogeneous datasets, including audio clips labeled with varying annotation granularity and with different sets of possible events. We propose a multi-iteration, multi-stage procedure for fine-tuning Audio Spectrogram Transformers on the joint *DESED* and *MAESTRO Real* datasets. The first stage closely matches the baseline system setup and trains a CRNN model while keeping the pre-trained transformer model frozen. In the second stage, both CRNN and transformer are fine-tuned using heavily weighted self-supervised losses. After the second stage, we compute strong pseudo-labels for all audio clips in the training set using an ensemble of fine-tuned transformers. Then, in a second iteration, we repeat the two-stage training process and include a distillation loss based on the pseudo-labels, achieving a new single-model, state-of-the-art performance on the public evaluation set of DESED with a PSDS1 of 0.692. A single model and an ensemble, both based on our proposed training procedure, ranked first in Task 4 of the DCASE Challenge 2024.[1].

Index Terms— DCASE Challenge, Sound Event Detection, ATST, BEATs, PaSST, DESED, MAESTRO Real, pseudo-labels

1. INTRODUCTION

The goal of Sound Event Detection (SED) is to identify specific acoustic events and their timing within audio recordings. Reliable SED systems enable applications in numerous domains, for example, in security and surveillance [1], smart homes [2], or health monitoring [3]. A main driver of research in this field is the annual DCASE Challenge, particularly Task 4, which focuses on SED.

State-of-the-art SED systems are based on deep learning approaches, requiring a substantial amount of annotated data. Their performance is mainly limited by the lack of strongly annotated real-world sound event datasets [4]. Hence, previous editions of Task 4 focused on learning from weakly labeled data [5], semi-supervised learning strategies [6], and utilizing synthetic strongly labeled data [7]. While Task 4 has been based on the DESED dataset [7] in previous years, the key novelty of the 2024 edition is a unified setup including a second dataset, MAESTRO Real [4]. As domain identification is prohibited, the goal is to develop a single system that can handle both datasets despite crucial differences, such as labels with different temporal granularity and potentially missing labels. In fact, because of the lack of strongly annotated, high-quality real-world data, the hope is that learning from two datasets in parallel has a synergetic effect and eventually increases the generalization performance on both datasets.

The main contributions of this work are as follows: **(1)** We introduce a multi-iteration, multi-stage training routine for fine-tuning pre-trained transformer models on SED using heterogeneous datasets. **(2)** We demonstrate that combining fine-tuned transformers – ATST [8], PaSST [9], and BEATs [10] – into a diverse ensemble to generate pseudo-labels, and using these pseudo-labels in a subsequent training iteration, significantly enhances single-model performance, yielding a relative increase of 25.9% in terms of polyphonic sound detection score [11,12] (PSDS1) on DESED and 2.7% in terms of segment-based mean partial area under the ROC curve (mpAUC) on MAESTRO, compared to the baseline system. **(3)** We conduct an ablation study to analyze the impact of the heterogeneous datasets and design choices related to them.

On DESED, we set a new state of the art on the public evaluation set, raising single-model performance from 0.686 [11] to 0.692 in terms of PSDS1. A single model and an ensemble, both based on our proposed training procedure, ranked first in the respective categories in Task 4 of the DCASE Challenge 2024 [13].

2. RELATED WORK

SED Architectures: Since the 2018 edition [14], the baseline system is based on a Convolutional Recurrent Neural Network (CRNN). A large increase in performance happened in the 2023 edition, as the baseline used BEATs [10] embeddings concatenated with CNN embeddings, which were then fed to the RNN, with a relative increase of almost 50% in PSDS1 score. Top-ranked systems in the 2023 edition improved over the baseline architecture with variations of frequency-dynamic convolution [15]. Recently, Shao et al. [16] proposed a procedure to fine-tune large pre-trained transformers on the DESED dataset with a two-stage training procedure, establishing a new state of the art. They showed that the key to avoiding overfitting is placing a large weight on the self-supervised losses to take advantage of the larger amount of unlabeled data.

Data Augmentation: As strongly annotated data is very limited, data augmentation is an important strategy to improve the generalization of SED systems. In this regard, Filter-Augment [17] simulates variations in the acoustic environment by applying different weights to frequency bands, forcing the model to extract information from wider frequency regions. Strategies for recording device generalization in Acoustic Scene Classification apply similar frequency weighting mechanisms: Frequency-MixStyle [18,19] mixes the frequency information of two audio clips in the dataset, and Device-Impulse augmentation [20] convolves an audio clip with an impulse response of a real recording device. Recently, Fre-

[1] Code: `https://github.com/CPJKU/cpjku_dcase24`

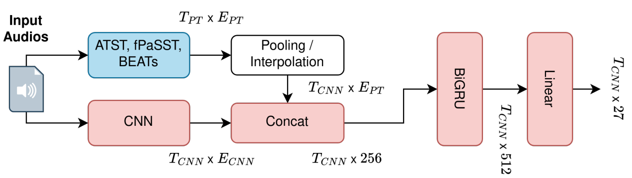


Figure 1: Overview of System Architecture. Blue are the pre-trained transformers; the red blocks together comprise the CRNN.

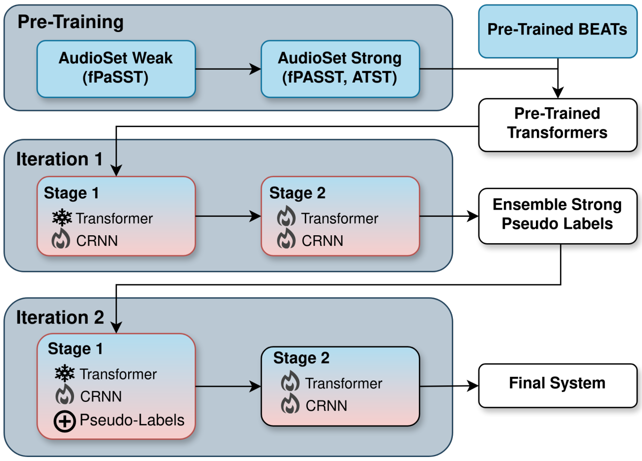


Figure 2: Training Pipeline. The snow flake symbol denotes frozen parameters, the flame that a model is trained in a particular stage.

quency Warping [8], which stretches or squeezes spectrograms in the frequency dimension, was shown to be an integral part when fine-tuning transformers on the DESED dataset [16].

Pseudo-labels: Both of the top-ranked approaches in the 2022 and 2023 editions of Task 4 computed pseudo-labels. Ebbers et al. [21] use a multi-iteration self-training procedure in which pseudo-labels, predicted by an ensemble, are iteratively refined. Kim et al. [22] employ a two-iterations setup in which strong pseudo-labels for weakly labeled, unlabeled, and AudioSet [23] clips are computed from an ensemble of models from the first training iteration. The computed pseudo-labels are converted into hard labels and used as additional targets in a second training iteration.

3. SYSTEM ARCHITECTURE

Figure 1 gives an overview of our SED system. It consists of two independent audio embedding branches (CNN and transformer), the outputs of which are pooled to the same sequence length. A Recurrent Neural Network (RNN) derives strong predictions from these combined sequences. Compared to the baseline [13] we experiment with two additional Audio Spectrogram Transformers besides BEATs [10], namely, ATST [8] and PaSST [9]. In addition to adaptive average pooling, we experiment with linear and nearest-exact interpolation to align transformer and CNN sequence lengths. In the following, we briefly describe the transformer models used in our setup. We refer the reader to [24] for more details.

ATST-Frame [25](denoted ATST in the following) was specifically designed to produce a sequence of frame-level audio embeddings. The architecture of ATST is based on the Audio Spectrogram Transformer (AST) [26]; it is pre-trained in a self-supervised manner on AudioSet. In our experiments, we use a checkpoint of ATST that is fine-tuned on the weak labels of AudioSet.

fPaSST: The Patchout faSt Spectrogram Transformer (PaSST) [9] is an improved version of the original AST [26] that shortens the training time and improves the performance via patchout regularization. We adapt PaSST to return frame-level predictions and call the resulting model Frame-PaSST (fPaSST). We pre-train fPaSST on the weakly annotated AudioSet using Knowledge Distillation [27], obtaining a mAP of 0.484.

BEATs: Likewise, BEATs [10] is also based on the AST [26] architecture; it was trained in an iterative, self-supervised procedure on AudioSet, where the BEATs encoder and tokenizer were alternately updated. In our experiments, we rely on the checkpoint of BEATs after the third iteration, where both the tokenizer and the encoder were fine-tuned on the weak labels of AudioSet.

4. TRAINING PIPELINE

In this section, we describe the pre-training routine on AudioSet strong and how the pre-trained models are fine-tuned on the Task 4 datasets in the proposed multi-iteration, multi-stage training procedure. An overview of the full training pipeline is shown in Figure 2. The full system architecture, depicted in Figure 1, is involved in all iterations and stages of Figure 2. The pre-trained transformers (blue block in Figure 1) are used as frozen audio embedding models in Stage 1 and fine-tuned together with the CRNN (red blocks in Figure 1) in Stage 2. The pseudo-labels are generated from an ensemble after Iteration 1 and used as additional prediction targets in Stage 1 of Iteration 2. In the following, we abbreviate Iteration {1,2} and Stage {1,2} as *I*{*1,2*} and *S*{*1,2*}, respectively.

4.1. Pre-Training on AudioSet strong

We hypothesize that the transformer models would benefit from additional pre-training on a large dataset strongly annotated for various acoustic events. To this end, we add a BiGRU block with 1024 units that processes the output of the transformer. We pre-train for 10 epochs on AudioSet strong [28], a subset of AudioSet that holds around 86,000 strongly labeled examples with annotations for 456 event classes. While ATST and, in particular, fPaSST benefit from this pre-training, there was no effect on the downstream task performance of BEATs; therefore, we only pre-train ATST and fPaSST on AudioSet strong. We select the checkpoint with the highest PSDS1 score on the AudioSet strong validation set for downstream training.

4.2. Multi-Stage Training

Inspired by [16], *I1* and *I2* are both split into two training stages. In *S1*, the CRNN (CNN + BiGRU) is trained from scratch while the large transformer model is frozen. This setup corresponds to the training of the baseline system with slightly different hyperparameters and additional data augmentations (as shown in Table 1).

In *S2*, the CRNN is initialized with pre-trained weights from *S1*, and both the CRNN and the transformer model are fine-tuned. As the system already performs well in its initial state, the transformer can rely on a high-quality self-supervised loss computed on the larger unlabeled set. Aligned with [16], in *S2*, we compute the interpolation consistency (ICT) loss [29] in addition to the mean teacher (MT) loss [30]. In both stages, we choose the best model based on the validation set by computing the sum of PSDS1 on the strongly labeled synthetic data, PSDS1 on external strongly labeled real data, and mpAUC on the MEASTRO validation set.

4.3. Multi-Iteration Training

After completing *I1*, we build an ensemble (see *Ensemble Stage 2* in Table 2) of multiple ATST, fPaSST, and BEATs models. This ensemble is used to compute strong pseudo-labels for all audio clips in the training set by averaging the frame-wise logits of the individual

Aug. Method	**Target**	**HP**	**Pipeline**
DIR [20]	All	p=0.5	*I*{*1,2*}.*S2*
Wavmix [33]	Str.	p=0.5,α=0.2	*I*{*1,2*}.*S*{*1,2*}
Freq-MixStyle [18]	All	p=0.5,α=0.3	*I1.S*{*1,2*},*I2.S2*
Mixup [33]	All	p=0.5,α=0.2	*I*{*1,2*}.*S*{*1,2*}
Time-Masking	DES. Str.	s=[0.05,0.3]	*I*{*1,2*}.*S2*
FilterAugment [17]	All	linear,p=0.8	*I1.S*{*1,2*},*I2.S2*
Freq-Warping [8]	All	p=0.5	*I*{*1,2*}.*S2*

Table 1: The table lists data augmentation methods, the data subset they are applied to (**Target**), hyperparameters (**HP**), and the respective iteration and stage they are used in (**Pipeline**). p is the probability for applying the augmentation method; α parameterizes Beta distributions; s specifies the masking ratio interval; and (*DES.*) *Str.* refers to strongly annotated audio clips (from DESED).

models. In *S1* of *I2*, we then use the pseudo-labels as additional prediction targets. We found that BCE is superior to MSE for computing the pseudo-label loss, and interestingly, using the pseudo-label loss only improves performance in *S1* of *I2* (see Table 4).

5. EXPERIMENTAL SETUP

5.1. Audio Pre-processing and Augmentation

For all models, we convert audio clips to 10 seconds in length at a 16 kHz sampling rate. For the CNN, we match the baseline settings and compute Mel spectrograms with 128 Mel bins using a window length of 128 ms and hop size of 16 ms. For the transformers, we use their original feature extraction pipelines [9, 10, 25].

Table 1 details all the data augmentation methods used in our training pipeline. In contrast to the baseline, we apply Cross-Dataset Mixup and Cross-Dataset Freq-MixStyle. That is, we mix audio clips from MAESTRO and DESED instead of keeping them separate. In the case of Mixup, we allow the loss to be calculated for all partially active classes, irrespective of the audio clip's dataset origin (see Section 5.3). For Wavmix and Mixup, we mix the pseudo-labels accordingly.

5.2. Datasets and Optimization

We use the DESED [7] and MAESTRO [4] datasets as provided for Task 4 in the DCASE 2024 challenge and, additionally, approximately 7,000 strongly annotated clips extracted from AudioSet strong according to [31]. We refer the reader to [24] for a detailed description of the data setup.

The training data can be seen as the union of five subsets: MAESTRO strong and DESED: real strong, synthetic strong, weakly annotated, and unlabeled. We draw batches of (12, 10, 10, 20, 20) and (56, 40, 40, 72, 72) samples from these datasets in *S1* and *S2*, respectively. The model is trained to minimize BCE loss on all (pseudo-)labeled audio clips and MSE loss for the self-supervised MT [30] and ICT [29]methods. We compute a weighted sum of all losses and tune the individual weights for all iterations and stages. AdamW [32] with weight decays of 1e-2 and 1e-3 is used in *S1* and *S2*, respectively. Learning rates are listed in Table 2.

5.3. Handling Heterogeneous Sound Event Classes

The DESED and MAESTRO datasets are annotated with two different sets of sound event classes. We adopt the baseline [13] strategy, in which the loss for an audio clip is calculated only on the dataset-specific event classes and mapped event classes, as explained in the following: To exploit the fact that the DESED and MAESTRO classes are not fully disjoint but partly represent the same concepts, the baseline introduces class mappings. For example, when the classes *people talking*, *children voices*, or *announcement* are active in a MAESTRO clip, the corresponding DESED class *Speech* is set to the same confidence value. In addition, we also include a mapping from MAESTRO to DESED classes. Specifically, we set the values of the MAESTRO classes *cutlery and dishes* and *people talking* to 1 if the DESED classes *Dishes* and *Speech* are present. This is also performed for weak class labels.

5.4. Postprocessing

For model selection and hyperparameter tuning, we stick with the same class-wise median filter used in the baseline system [13]. After selecting the best configurations for each model, we apply the recently introduced Sound Event Bounding Boxes (SEBBs) [34] method for postprocessing. We use class-wise parameters and tune them by using linearly spaced search grids (8 values) for step filter length (0.38 to 0.66), relative merge threshold (1.5 to 3.25), and absolute merge threshold (0.15 to 0.325).

6. RESULTS

In this section, we present the results of the described models (Section 3) in the introduced training pipeline (Section 4). Table 2 lists the best configuration and the corresponding results on the test set for each architecture in both iterations and stages. The table lists the sequence pooling method (**Seq.**) and the CNN (**lr_cnn**), RNN (**lr_rnn**), and Transformer (**lr_tf**) learning rates. **lr_dec** indicates the layer-wise learning rate decay for the transformers as used in [16].

In *I1.S1*, in which the transformers are frozen, BEATs seems to extract the embeddings of the highest quality, followed by fPaSST and ATST. *I1.S1* with BEATs is very similar to the baseline [13] and achieves a similar rank score with a slight performance increase in our setup. Compared to *I1.S1*, all three transformers demonstrate a large increase in rank score when fine-tuned on the Task 4 datasets in *I1.S2*. Notably, the three transformers have different strengths, with ATST and BEATs achieving the best scores on MAESTRO and DESED clips, respectively. *Ensemble Stage 2* denotes an ensemble of 46 models resulting from *I1.S2*, including ATST, fPaSST, and BEATs trained in different configurations. We use *Ensemble Stage 2* to generate strong pseudo-labels for all audio clips in the dataset.

The additional pseudo-label loss in *I2.S1* boosts performance substantially, with all three transformers achieving a higher rank score compared to *I1.S2*. The top rank scores for all models are achieved in *I2.S2*, with ATST obtaining the highest rank score.

Table 3 presents the top configurations of ATST, fPaSST, and BEATs from *I2.S2* with the state-of-the-art postprocessing method cSEBBs [34] applied. *ATST* and *ATST DT*, a variant of ATST that is trained on all available audio clips included in the Task 4 development set, were submitted as single models to the challenge. *ATST DT* using cSEBBs postprocessing achieves a PSDS1 of 0.692 on the public evaluation set of DESED, improving over the previous state of the art (0.686 PSDS1) [34].

6.1. Ablation Study

Table 4 shows the results of ATST for *I2.S1* and *I2.S2* trained in different configurations to analyze the design choices related to the heterogeneous datasets and the pseudo-label loss. For settings - *DESED* and - *MAESTRO*, the proposed system is trained only on MAESTRO and DESED data, respectively. We find that training on

		Model	lr_cnn	lr_rnn	lr_tf	lr_dec	Seq.	mpAUC	PSDS1	Rank Score
Iteration 1	Stage 1	ATST	1e-3	1e-3	-	-	int. lin.	0.702 ± 0.008	0.493 ± 0.012	1.195 ± 0.012
		fPaSST	1e-3	1e-3	-	-	int. nearest	0.709 ± 0.021	0.502 ± 0.010	1.212 ± 0.027
		BEATs	1e-3	1e-3	-	-	int. nearest	0.719 ± 0.004	0.509 ± 0.003	**1.228 ± 0.006**
	Stage 2	ATST	1e-4	1e-3	1e-4	0.5	int. nearest	0.739 ± 0.017	0.520 ± 0.005	**1.259** ± 0.020
		fPaSST	1e-4	1e-3	1e-4	1	int. nearest	0.726 ± 0.021	0.514 ± 0.008	1.24 ± 0.027
		BEATs	1e-4	1e-3	1e-4	1	int. lin.	0.713 ± 0.002	0.539 ± 0.004	1.252 ± 0.003
	Ensemble Stage 2		-	-	-	-	mix	0.735	0.569	1.303
Iteration 2	Stage 1	ATST	5e-4	5e-4	-	-	avg. pool	0.741 ± 0.017	0.536 ± 0.006	**1.277 ± 0.012**
		fPaSST	5e-4	5e-4	-	-	int. nearest	0.722 ± 0.011	0.526 ± 0.004	1.248 ± 0.012
		BEATs	5e-4	5e-4	-	-	int. nearest	0.724 ± 0.011	0.537 ± 0.005	1.262 ± 0.010
	Stage 2	ATST	1e-5	1e-4	1e-4	0.5	avg. pool	0.750 ± 0.004	0.548 ± 0.004	**1.298** ± 0.006
		fPaSST	5e-5	5e-4	1e-4	1	int. nearest	0.719 ± 0.013	0.539 ± 0.003	1.259 ± 0.015
		BEATs	5e-5	5e-4	1e-4	1	int. nearest	0.729 ± 0.005	0.557 ± 0.005	1.286 ± 0.009

Table 2: The table presents the results of ATST, fPaSST, and BEATs for both iterations and stages on the official development test set. For each model, we list the best configuration in terms of the sequence length adaptation method (**Seq.**), where *int. lin.*, *int. nearest*, *avg. pool*, and *mix* denote linear and nearest-exact interpolation, adaptive average pooling, and a mixture of these methods, respectively. *Ensemble Stage 2* is used to generate the pseudo-labels for Iteration 2. **Rank Score** denotes the sum of **mpAUC** and **PSDS1**.

Model	mpAUC	PSDS1 MF	PSDS1*	Ev. PSDS1*
ATST	0.750	0.548	0.617	0.684
fPaSST	0.719	0.539	0.601	0.681
BEATs	0.729	0.557	0.622	0.683
ATST DT	✗	✗	✗	0.692

Table 3: Results for best single-model configurations of ATST, fPaSST, and BEATs from *I2.S2*. **PSDS1** lists results with a median filter; **PSDS1*** results using cSEBBs postprocessing [34]; and **Ev. PSDS1*** lists results on the DESED public evaluation set with cSEBBs postprocessing. *ATST DT* denotes the best ATST configuration trained on the full development set.

DESED and MAESTRO simultaneously is beneficial for the performance on both datasets, which coincides with the finding reported for the baseline system [13]. For both stages of *I2*, excluding MAESTRO clips when calculating the self-supervised losses (- *SSL MAESTRO*) and not mapping event classes from MAESTRO to DESED (- *MAESTRO-DESED Map.*, see Section 5.3) leads to a performance decrease. However, we find no clear answer to the question of whether the SSL loss should be calculated on all classes or only on the dataset-specific classes of an audio clip (*+/- SSL class mask*); *S1* and *S2* benefit from different settings. Interestingly, using the pseudo-label loss in *I2.S2* (+ *Pseudo Loss*) does not increase the rank score. Therefore, the setup in *I1.S2* and *I2.S2* remains identical, which demonstrates that a well-trained CRNN from *S1* can have a large impact on the performance achieved in *S2*. We also tried to use separate heads for predictions on DESED and MAESTRO classes and realized this with an additional single BiGRU layer per dataset (+ *Separate RNN layer*), which resulted in a performance decrease. Further obvious choices, such as thresholding the pseudo-labels by 0.5 (+ *Hard Pseudo*) and calculating the pseudo-label loss on all classes (+ *Pseudo All Classes*) instead of only dataset-specific classes, are inferior to our proposed strategy as well.

System	mpAUC	PSDS1	Rank Score
ATST *I2.S1*	0.741	0.536	**1.277**
- DESED	0.724	-	-
- MAESTRO	-	0.531	-
- SSL MAESTRO	0.741	0.535	1.276
- MAESTRO-DESED Map.	0.717	0.530	1.247
+ SSL class mask	0.740	0.530	1.27
+ Separate RNN layer	0.714	0.531	1.244
+ Hard Pseudo	0.706	0.538	1.244
+ Pseudo All Classes	0.717	0.534	1.25
ATST *I2.S2*	0.750	0.548	**1.298**
- SSL MAESTRO	0.743	0.546	1.289
- MAESTRO-DESED Map.	0.749	0.547	1.297
- SSL class mask	0.749	0.544	1.293
+ Pseudo Loss	0.746	0.552	1.297

Table 4: Ablation Study on design choices related to the heterogeneous datasets and the pseudo-label loss used in *I2.S1*. The study is performed on the top single model, ATST, trained in *I2.S1* (upper part) and in *I2.S2* (lower part).

7. CONCLUSION

This paper presented a multi-iteration, multi-stage training routine for fine-tuning transformers on the SED task with heterogeneous datasets. We showed that the performance of all tested systems monotonously increases throughout both iterations and stages. The proposed method led to a new state-of-the-art performance of 0.692 in PSDS1 on the DESED public evaluation set and achieved the top rank in Task 4 of the DCASE 2024 challenge. We specifically studied design choices related to the heterogeneous datasets and found that simultaneously training on DESED and MAESTRO leads to a performance increase on both datasets compared to training the system on a single dataset.

8. ACKNOWLEDGMENT

The computational results presented were achieved in part using the Vienna Scientific Cluster (VSC) and the Linz Institute of Technology (LIT) AI Lab Cluster. The LIT AI Lab is supported by the Federal State of Upper Austria. Gerhard Widmer's work is supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme, grant agreement No 101019375 (Whither Music?).

9. REFERENCES

[1] R. Radhakrishnan, A. Divakaran, and A. Smaragdis, "Audio analysis for surveillance applications," in *WASPAA*, 2005.

[2] C. Debes, A. Merentitis, S. Sukhanov, M. E. Niessen, N. Frangiadakis, and A. Bauer, "Monitoring activities of daily living in smart homes: Understanding human behavior," *IEEE Signal Process. Mag.*, 2016.

[3] Y. Zigel, D. Litvak, and I. Gannot, "A method for automatic fall detection of elderly people using floor vibrations and sound - proof of concept on human mimicking doll falls," *IEEE Trans. Biomed. Eng.*, 2009.

[4] I. Martín-Morató and A. Mesaros, "Strong labeling of sound events using crowdsourced weak labels and annotator competence estimation," *IEEE ACM Trans. Audio Speech Lang. Process.*, 2023.

[5] Q. Kong, Y. Xu, W. Wang, and M. D. Plumbley, "Sound event detection of weakly labelled data with cnn-transformer and automatic threshold optimization," *IEEE ACM Trans. Audio Speech Lang. Process.*, 2020.

[6] S. Park, A. Bellur, D. K. Han, and M. Elhilali, "Self-training for sound event detection in audio mixtures," in *ICASSP*, 2021.

[7] N. Turpault, R. Serizel, A. Parag Shah, and J. Salamon, "Sound event detection in domestic environments with weakly labeled data and soundscape synthesis," in *DCASE Workshop*, 2019.

[8] X. Li and X. Li, "ATST: audio representation learning with teacher-student transformer," in *Interspeech*, 2022.

[9] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, "Efficient training of audio transformers with patchout," in *Interspeech*, 2022.

[10] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, Z. Chen, W. Che, X. Yu, and F. Wei, "Beats: Audio pre-training with acoustic tokenizers," in *ICML*, 2023.

[11] J. Ebbers, R. Haeb-Umbach, and R. Serizel, "Threshold independent evaluation of sound event detection scores," in *ICASSP*, 2022.

[12] Ç. Bilen, G. Ferroni, F. Tuveri, J. Azcarreta, and S. Krstulovic, "A framework for the robust evaluation of sound event detection," in *ICASSP*, 2020.

[13] S. Cornell, J. Ebbers, C. Douwes, I. Martín-Morató, M. Harju, A. Mesaros, and R. Serizel, "DCASE 2024 task 4: Sound event detection with heterogeneous data and missing labels," *CoRR*, 2024.

[14] R. Serizel, N. Turpault, H. Eghbal-Zadeh, and A. P. Shah, "Large-scale weakly labeled semi-supervised sound event detection in domestic environments," in *DCASE Workshop*, 2018.

[15] H. Nam, S. Kim, B. Ko, and Y. Park, "Frequency dynamic convolution: Frequency-adaptive pattern recognition for sound event detection," in *Interspeech*, 2022.

[16] N. Shao, X. Li, and X. Li, "Fine-tune the pretrained atst model for sound event detection," in *ICASSP*, 2024.

[17] H. Nam, S. Kim, and Y. Park, "Filteraugment: An acoustic environmental data augmentation method," in *ICASSP*, 2022.

[18] B. Kim, S. Yang, J. Kim, H. Park, J. Lee, and S. Chang, "Domain generalization with relaxed instance frequency-wise normalization for multi-device acoustic scene classification," in *Interspeech*, 2022.

[19] F. Schmid, S. Masoudian, K. Koutini, and G. Widmer, "Knowledge distillation from transformers for low-complexity acoustic scene classification," in *DCASE Workshop*, 2022.

[20] T. Morocutti, F. Schmid, K. Koutini, and G. Widmer, "Device-robust acoustic scene classification via impulse response augmentation," in *EUSIPCO*, 2023.

[21] J. Ebbers and R. Haeb-Umbach, "Pre-training and self-training for sound event detection in domestic environments," DCASE Challenge, Tech. Rep., 2022.

[22] J. W. Kim, S. W. Son, Y. Song, H. K. Kim, I. H. Song, and J. E. Lim, "Semi-supervsied learning-based sound event detection using freuqency dynamic convolution with large kernel attention for DCASE challenge 2023 Task 4," DCASE Challenge, Tech. Rep., 2023.

[23] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *ICASSP*, 2017.

[24] F. Schmid, P. Primus, T. Morocutti, J. Greif, and G. Widmer, "Improving audio spectrogram transformers for sound event detection through multi-stage training," DCASE Challenge, Tech. Rep., 2024.

[25] X. Li, N. Shao, and X. Li, "Self-supervised audio teacher-student transformer for both clip-level and frame-level tasks," *IEEE ACM Trans. Audio Speech Lang. Process.*, 2024.

[26] Y. Gong, Y. Chung, and J. R. Glass, "AST: audio spectrogram transformer," in *Interspeech*, 2021.

[27] F. Schmid, K. Koutini, and G. Widmer, "Efficient large-scale audio tagging via transformer-to-cnn knowledge distillation," in *ICASSP*, 2023.

[28] S. Hershey, D. P. W. Ellis, E. Fonseca, A. Jansen, C. Liu, R. C. Moore, and M. Plakal, "The benefit of temporally-strong labels in audio event classification," in *ICASSP*, 2021.

[29] X. Zheng, Y. Song, I. McLoughlin, L. Liu, and L. Dai, "An improved mean teacher based method for large scale weakly labeled semi-supervised sound event detection," in *ICASSP*, 2021.

[30] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *ICLR Workshop*, 2017.

[31] S. Xiao, J. Shen, A. Hu, X. Zhang, P. Zhang, and P. Yan, "Sound event detection with weak prediction for dcase 2023 challenge task4a," DCASE Challenge, Tech. Rep., 2023.

[32] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *ICLR*, 2019.

[33] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *ICLR*, 2018.

[34] J. Ebbers, F. G. Germain, G. Wichern, and J. L. Roux, "Sound event bounding boxes," *accepted at Interspeech*, 2024.

REPRESENTATIONAL LEARNING FOR AN ANOMALOUS SOUND DETECTION SYSTEM WITH SOURCE SEPARATION MODEL

Seunghyeon Shin[1], *Seokjin Lee*[1,2],

[1] School of Electronic and Electrical Engineering, Kyungpook National University,
Daegu, Republic of Korea, {sh.shin, sjlee6}@knu.ac.kr
[2] School of Electronics Engineering, Kyungpook National University, Daegu, Republic of Korea

ABSTRACT

The detection of anomalous sounds in machinery operation presents a significant challenge due to the difficulty in generalizing anomalous acoustic patterns. This task is typically approached as an unsupervised learning or novelty detection problem, given the complexities associated with the acquisition of comprehensive anomalous acoustic data. Conventional methodologies for training anomalous sound detection systems primarily employ auto-encoder architectures or representational learning with auxiliary tasks. However, both approaches have inherent limitations. Auto-encoder structures are constrained to utilizing only the target machine's operational sounds, while training with auxiliary tasks, although capable of incorporating diverse acoustic inputs, may yield representations that lack correlation with the characteristic acoustic signatures of anomalous conditions. We propose a training method based on the source separation model (CMGAN[1]) that aims to isolate non-target machine sounds from a mixture of target and non-target class acoustic signals. This approach enables the effective utilization of diverse machine sounds and facilitates the training of complex neural network architectures with limited sample sizes. Our experimental results demonstrate that the proposed method yields better performance compared to both conventional auto-encoder training approaches and source separation techniques that focus on isolating target machine signals. Moreover, our experimental results demonstrate that the proposed method exhibits the potential for enhanced representation learning as the quantity of non-target data increases, even while maintaining a constant volume of target class data.

Index Terms— Anomalous sound detection, Novelty detection, Representational learning, Source separation

1. INTRODUCTION

Anomalous sound detection aims to determine whether an acquired acoustic signal originates from normal or anomalous operating conditions. The diverse nature of anomalous sounds, which vary depending on the target system, necessitates separate datasets for each target system. Furthermore, the difficulty in acquiring comprehensive anomalous sound samples, coupled with the fact that acquired samples may not represent all possible anomalous states, necessitates approaching this problem as an unsupervised learning or novelty detection task.

In response to these challenges, DCASE 2024 Task 2 [2] requires the development of an anomalous sound detection model capable of monitoring machine conditions without access to anomalous samples. The task provides 1,000 sound clips per machine type for model training. Additionally, to ensure generalization across varying operating environments and machine types, the system must be robust to environmental changes and operate without reliance on additional attribute information, such as operating speed or machine identification.

Conventional approaches typically employ neural networks to extract representations from acoustic signals. These methods can be broadly categorized into two main groups. The first utilizes auto-encoder structures [3], training neural networks to encode and decode input signals. Anomaly detection is then performed either by quantifying the discrepancy between the decoded output and the input, or by using the encoder output as an embedding feature for subsequent anomaly score calculation. The second category involves training neural networks on auxiliary tasks, such as classifying additional data attributes or classes [4], or directly learning representations through contrastive learning techniques [5].

Although source separation neural networks have been used in anomalous sound detection [6], they have mainly served as preprocessing stages for other neural networks or have been trained using attribute information. To address the constraints of training without attribute information while effectively utilizing non-target class signals, we propose a novel source separation-based representational learning method.

Our experimental results demonstrate the efficacy of the proposed method. We conducted comparative tests against conventional source separation methods that estimate target class signals and auto-encoder structures that utilize only the target signal as input. The proposed method achieved a better representation when evaluated by using anomalous sound detection with the Mahalanobis distance. Moreover, we observed that the performance of our proposed method improves with increased non-target data, even when the quantity of target data remains constant.

2. METHODOLOGY

2.1. Training strategy and anomaly score calculation

We obtain an embedding feature vector from a source separation model. The purpose of the neural network is to extract characteristics that can distinguish whether a machine's condition is normal or abnormal. We assume that normal and abnormal conditions can be distinguished from acoustic data and that would also be distinguishable in the representation of neural network. We utilized the CMGAN[1] neural network structure, which is an encoder-decoder structure with conformer blocks[7].

In contrast to a typical auto-encoder structure, where the neural network is trained to reconstruct the desired input signal at the output, our training objective is to remove the target machine signal from the input and separate the signals from other machines. In our representation training process, the input of the neural network $X_{t,f}$ is expressed as follows:

$$X_{t,f} = \mathcal{F}(d_c(t) + s \times n_{\bar{c}}(t)), \quad (1)$$

where $d_c(t)$ represents the signal of the target machine class c signal in time series, $n_{\bar{c}}(t)$ represents the signal from another class $\bar{c}$ that is not the target machine class, s is the scaling factor to match the intensity of the target and other machine class signals in dB scale, and the neural network input $X_{t,f}$ in the time-frequency domain is obtained by applying the short-time Fourier transform operator $\mathcal{F}$ to the weighted sum of $d_c(t)$ and $n_{\bar{c}}(t)$. We introduce a scaling factor s to modulate the training objective's difficulty. Small value of s make the intensity of non-target signal smaller, the source separation problem becomes more difficult, and vice versa. For the proposed system to work well, s value needs to be sufficiently small, but a large s value may be needed when the number of training samples or model complexity is limited.

The neural network utilizes the real, imaginary, and magnitude components of the spectrogram as input. Since we intend for our feature extractor to remove the target machine signal, we trained it to minimize the difference between the neural network's estimation output and the other class signal. We configured the training loss function $\mathcal{L}$ of the neural network as follows:

$$\mathcal{L} = \alpha\{\frac{1}{l}\sum_{t=1}^{l}(|s \times n_{\bar{c}}(t) - y(t)|)\} + \beta\{\frac{1}{mn}\sum_{t=1}^{m}\sum_{f=1}^{n}(|s \times N^R_{t,f}| - |Y^R_{t,f}|)^2\} + \gamma\{\frac{1}{mn}\sum_{t=1}^{m}\sum_{f=1}^{n}(|s \times N^I_{t,f}| - |Y^I_{t,f}|)^2\}, \quad (2)$$

where y_t and $Y^R_{t,f}$, $Y^I_{t,f}$ are the output of the neural network decoder in the time series and the real and imaginary components of the time-frequency domain, respectively. $N^R_{t,f}$ and $N^I_{t,f}$ represent the real and imaginary components of the non-target signal in the time-frequency domain. α, β, and γ are the hyperparameters for each difference term. During the training procedure, the network is trained to estimate $n_{\bar{c}}(t)$ from $X_{t,f}$.

Anomaly scores are calculated from the Mahalanobis distance of the neural network output feature matrix. The covariance matrix of the feature is estimated by the maximum-likelihood covariance estimator. In summary, we first train the neural network to separate the signal from the mixed signal of other classes and the target class signal, excluding the target class signal. After training, the average pooling is performed in multiple stages of the network, and the average pool is executed from the output of the encoder, the intermediate of the conformer, and the output of the conformer. The resulting average-pooled matrices are then used as features for anomaly scoring, and in the scoring process, we employ the Mahalanobis distance with a covariance estimator. The overview of our system is shown in Fig. 1.

2.2. Experiments configure

We utilized two datasets consistent with those employed in DCASE 2024 Task 2: ToyADMOS2 [8] and MIMII DG [9]. These datasets collectively provided 16 types of machine sounds, each comprising 1,000 training clips. Each set of 1,000 clips was composed of 990 clips from the source domain and 10 clips from the target domain which are different operating environment from the source domain. For evaluation purposes, 7 machine type signals were provided, consisting of 200 sound clips labeled as either normal or anomalous. The evaluation data were equally distributed between the source and target domains, with 100 clips from each domain. The source domain, constituting 99% of the training data, represents the primary training environment. The target domain, comprising the remaining 1% of training data, consists of domain-shifted signals that reflect changed machine operating conditions relative to the source domain.

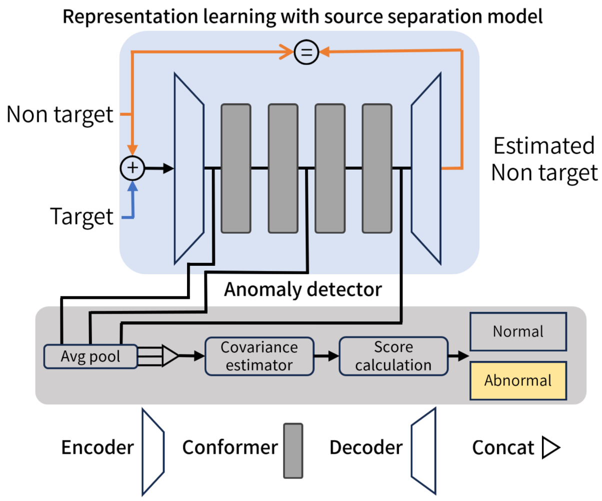


Figure 1: Proposed anomalous detection system overview

To assess the efficacy of our proposed method, we conducted experiments using varying quantities of non-target data. Two distinct datasets were employed: the first included non-target sounds from six machine types that contained test data, while the second utilized sounds from 14 machine types, excluding the Brushless Motor class due to the presence of clipping in some samples. Audio preprocessing involved randomly trimming each 16kHz sampled clip to 2 seconds, followed by a short-time Fourier transform using a filter length of 400 samples and an overlap of 100 samples. In the decibel matching process, non-target class signals were attenuated by 5dB lower relative to the target class signal. We added average pooling layers to the CMGAN neural network structure, applying pooling to each channel. Given the network's 64-channel architecture, the resulting feature vector maintained a 64-dimensional size post-pooling. The average pooling was performed at three locations: the encoder output, the second conformer block output, and the last conformer block output (decoder input). The 2-second average-pooled results were concatenated to form the final feature set. The loss function hyperparameters α, β, and γ were set to 0.5, 6.0, and 1.0, respectively, and remained constant across all machine classes. We initially adopted hyperparameters value from the original hyperparameters of CMGAN and subsequently fine-tuned them through experimentation. The hyperparameter values are dependent on the non-target signal intensity matching level and necessitating adjustment as matching dB changes. For optimization, we used the AdamW [10] algorithm in conjunction with a StepLR learning rate scheduler. To ensure fair comparison, hyperparameters and neural network configurations were maintained consistently across different training methodologies.

3. EVALUATION

3.1. Evaluation metric

The performance of the anomalous detection system is evaluated using the Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) curve. AUC scores are calculated in three distinct contexts: the source domain, which generates most of the training data; the target domain, representing a domain-shifted environment; and the partial AUC (pAUC), which constrains the maximum false positive rate to 10%, thus addressing the frequency of false alarms in practical applications. The overall model performance, consistent with the DCASE 2024 Task 2 official score Ω, is computed as the harmonic mean of results across all classes, domains, and the pAUC, as follows:

$$\Omega = h\{AUC_{c,d}, pAUC_c | c \in \mathcal{C}, d \in \{source, target\}\}, \quad (3)$$

where h is the harmonic mean operator and $\mathcal{C}$ is the set of the machine types.

3.2. Evaluation result

We evaluated our proposed method against two conventional approaches: a source separation method utilizing the CMGAN structure and an auto-encoder method also based on CMGAN. For comprehensive performance comparison, we have included the results of the DCASE 2024 Task 2 baseline system [11] in Table 1. Table 1 presents the performance of various systems. Our proposed method is evaluated with two configurations: estimating 14 and 6 non-target classes from a mixture of target and non-target signals. The conventional separation approach trains a neural network to estimate the target class from a mixed signal, while the auto-encoder method reconstructs the target signal from target-only input. The DCASE 2024 Task 2 baseline systems score anomalies using the mean square error (Baseline-MSE) and the Mahalanobis distance (Baseline-Mahalanobis) between input and output.

The score Ω, which represents the harmonic mean of the AUC and pAUC performance metrics, demonstrates the efficacy of our approach. The neural network trained to separate non-target signals achieved a Ω score of 54.58% This performance surpasses both the conventional separation method, which estimates target signals and achieved 53.99%, and the auto-encoder method, which yielded 51.41%. Furthermore, we observed that increasing the quantity of non-target class data enhanced the potential for acquiring better representations. Specifically, expanding the diversity of non-target classes led to an improvement in the Ω score from 54.58% to 56.00% when using our proposed training method. To visualize the effectiveness of our approach, we employed the t-Distributed Stochastic Neighbor Embedding (tSNE) projection to represent the learned features of the toytrain class, as illustrated in Fig. 2. The visualization demonstrates that our proposed method, which focuses on separating non-target class signals, yields more distinct separations between normal and anomalous samples compared to alternative methods. These alternatives, which include approaches trained to separate target class signals or traditional auto-encoder methods, show less clear differentiation in their projected representations.

4. CONCLUSIONS

This study proposes a novel approach to representation learning for anomalous detection systems, utilizing a neural network with a source separation model. Given the constraints of having only normal condition samples and limited training data, we developed a representation learning strategy that separates non-target class signals from a mixture of target and non-target class signals. Our method effectively leverages both the available training samples and data from other classes. To evaluate the neural network's ability to learn representations that distinguish anomalous characteristics, we implemented an anomalous detection system that scores the obtained representations using the Mahalanobis distance and a maximum-likelihood covariance estimator. We compared our proposed method with alternative training strategies, including separating target signals from mixed sounds and estimating target signals from target-only inputs. Results demonstrate that our method achieves superior performance, with a harmonic mean score of 54.58%, compared to 53.99% and 51.41% for the alternative approaches. Notably, we observed that our training strategy yields improved representations with an increase in non-target class signals, even when the quantity of target class signals remains constant. Specifically, utilizing 14 non-target classes resulted in a score of 56.00%, a 1.42% improvement over the 6-class non-target scenario, and better results compared to the baseline methods (55.35% and 55.02%) employing two different anomalous scoring techniques. Visualization of the learned representations using a t-SNE projection further corroborates the efficacy of our approach, revealing a more distinct separation between normal and anomalous samples compared to other methods. In conclusion, we have proposed and validated a training strategy that effectively utilizes both target and non-target class samples. Our method of training neural networks to separate non-target signals from mixed inputs demonstrates better performance in obtaining target class representations compared to target separation and auto-encoder methods. Furthermore, we have shown that our approach can achieve enhanced representations with an increased diversity of non-target class signals, highlighting its potential for scalability and improved performance in anomalous sound detection tasks.

5. REFERENCES

[1] S. Abdulatif, R. Cao, and B. Yang, "Cmgan: Conformer-based metric-gan for monaural speech enhancement," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 2477–2493, 2024.

[2] T. Nishida, N. Harada, D. Niizumi, D. Albertini, R. Sannino, S. Pradolini, F. Augusti, K. Imoto, K. Dohi, H. Purohit, T. Endo, and Y. Kawaguchi, "Description and discussion on DCASE 2024 challenge task 2: First-shot unsupervised anomalous sound detection for machine condition monitoring," *In arXiv e-prints: 2406.07250*, 2024.

[3] K. Li, Q.-H. Nguyen, Y. Ota, and M. Unoki, "Unsupervised anomalous sound detection for machine condition monitoring using temporal modulation features on gammatone auditory filterbank." in *DCASE*, 2022.

[4] S. Venkatesh, G. Wichern, A. S. Subramanian, and J. Le Roux, "Improved domain generalization via disentangled multi-task learning in unsupervised anomalous sound detection." in *DCASE*, 2022.

[5] X. Cai and H. Dinkel, "A contrastive semi-supervised learning framework for anomaly sound detection," in *Proceedings of the 6th Detection and Classification of Acoustic Scenes*

System info	Metric	bearing	fan	gearbox	slider	toycar	toytrain	valve	Ω score (h-mean)
Proposed method (14 class)	**AUC(Target)**	69.24%	62.76%	59.84%	55.88%	45.96%	65.64%	48.76%	56.00%
	AUC(Source)	61.04%	58.60%	68.68%	65.88%	44.72%	76.76%	46.36%	
	pAUC	58.05%	54.16%	55.05%	51.58%	48.89%	54.79%	48.89%	
Proposed method (6 class)	**AUC(Target)**	69.52%	63.28%	62.12%	47.16%	44.32%	65.36%	47.48%	54.58%
	AUC(Source)	61.04%	55.64%	61.84%	60.28%	47.68%	77.52%	48.08%	
	pAUC	54.16%	51.42%	52.84%	51.58%	48.68%	51.84%	48.79%	
Conventional separation	**AUC(Target)**	67.96%	57.00%	56.36%	53.72%	47.00%	65.08%	48.00%	53.99%
	AUC(Source)	64.52%	57.96%	61.80%	61.64%	42.48%	74.64%	42.24%	
	pAUC	56.32%	48.74%	51.79%	51.63%	47.84%	53.53%	48.63%	
Auto-encoder	**AUC(Target)**	67.84%	55.48%	49.84%	42.16%	49.72%	63.04%	47.56%	51.41%
	AUC(Source)	57.92%	57.92%	53.12%	44.92%	43.24%	76.76%	39.48%	
	pAUC	51.89%	51.42%	51.05%	51.05%	48.11%	53.00%	49.32%	
Baseline (MSE)	**AUC(Target)**	61.40%	55.24%	69.34%	56.01%	33.75%	46.92%	46.25%	55.35%
	AUC(Source)	62.01%	67.71%	70.40%	66.51%	66.98%	76.63%	51.07%	
	pAUC	57.58%	57.53%	55.65%	51.77%	48.77%	47.95%	52.42%	
Baseline (Mahalanobis)	**AUC(Target)**	51.58%	42.70%	74.35%	68.11%	37.35%	39.99%	53.61%	55.02%
	AUC(Source)	54.43%	79.37%	81.82%	75.35%	63.01%	61.99%	55.69%	
	pAUC	58.82%	53.44%	55.74%	49.05%	51.04%	48.21%	51.26%	

Table 1: Anomalous detection performance comparison of proposed method and others

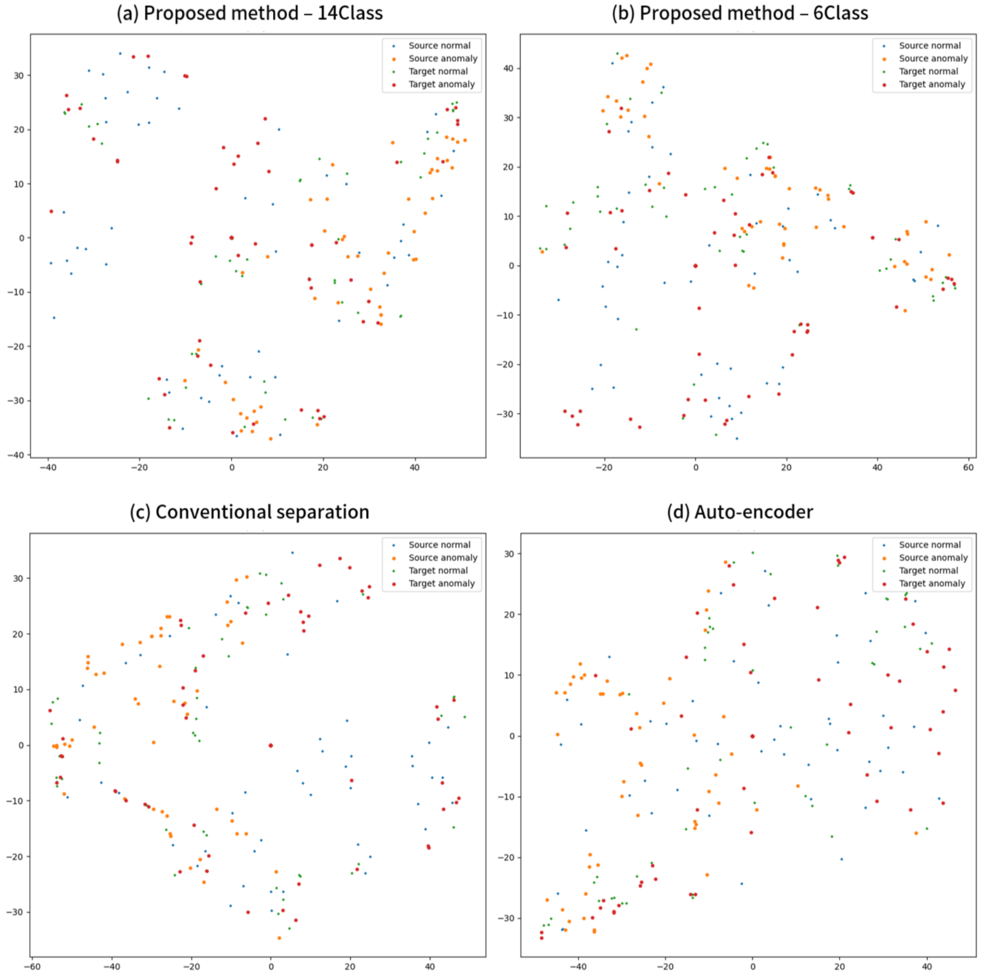


Figure 2: tSNE projection of representation of neural network trained by (a) Proposed method with 14 non-target classes (b) Proposed method with 6 non-target classes (c) Source separation trained to separate target class (d) Auto-encoder structure

and Events 2021 Workshop (DCASE2021), Barcelona, Spain, November 2021, pp. 31–34.

[6] K. Shimonishi, K. Dohi, and Y. Kawaguchi, "Anomalous Sound Detection Based on Sound Separation," in *Proc. INTERSPEECH 2023*, 2023, pp. 2733–2737.

[7] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.

[8] N. Harada, D. Niizumi, D. Takeuchi, Y. Ohishi, M. Yasuda, and S. Saito, "ToyADMOS2: Another dataset of miniature-machine operating sounds for anomalous sound detection under domain shift conditions," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, Barcelona, Spain, November 2021, pp. 1–5.

[9] K. Dohi, T. Nishida, H. Purohit, R. Tanabe, T. Endo, M. Yamamoto, Y. Nikaido, and Y. Kawaguchi, "MIMII DG: Sound dataset for malfunctioning industrial machine investigation and inspection for domain generalization task," in *Proceedings of the 7th Detection and Classification of Acoustic Scenes and Events 2022 Workshop (DCASE2022)*, Nancy, France, November 2022.

[10] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2019.

[11] N. Harada, D. Niizumi, Y. Ohishi, D. Takeuchi, and M. Yasuda, "First-shot anomaly sound detection for machine condition monitoring: A domain generalization baseline," in *2023 31st European Signal Processing Conference (EUSIPCO)*, 2023, pp. 191–195.

AUXILIARY DECODER-BASED LEARNING OF SOUND EVENT DETECTION USING MULTI-CHANNEL FEATURES AND MAXIMUM PROBABILITY AGGREGATION

Sang Won Son[1], Jongyeon Park[1], Hong Kook Kim[1,2]

[1] AI Graduate School
[2] School of EECS
Gwangju Institute of Science and Technology
Gwangju 61005, Korea
{{ssw970519, jypark3737}@gm., hongkook@}gist.ac.kr

Sulaiman Vesal[3]

[3]AI Lab., Innovation Center
Hanwha Vision
Teaneck, NJ 07666, USA
s.vesal@hanwha.com

Jeong Eun Lim[4]

[4]AI Lab., R&D Center
Hanwha Vision
Seongnam-si, Gyeonggi-do 13488, Korea
je04.lim@hanwha.com

ABSTRACT

This paper proposes a sound event detection (SED) model operating on heterogeneous labeled and/or unlabeled datasets, such as the DESED and MAESTRO datasets. The proposed SED model is based on a frequency dynamic convolution (FDY)–large kernel attention (LKA)-convolutional recurrent neural network (CRNN), and it is trained via mean-teacher-based semi-supervised learning to handle unlabeled data. The FDY–LKA-CRNN model incorporates bidirectional encoder representation from audio transformer (BEATs) embeddings to improve high-level semantic representation. However, the contribution of the BEATs encoder to the performance of the combined SED model is over-emphasized relative to that of the FDY–LKA-CRNN, which limits the overall performance of the SED model. To prevent this problem, an auxiliary decoder is applied to train the SED model with BEATs embeddings. Additionally, to accommodate the different recording characteristics of sound events in the two datasets, multi-channel log-mel features are concatenated in a channel-wise manner. Finally, a maximum probability aggregation (MPA) approach is proposed to address the different labeling time intervals of the two datasets. The performance of the proposed SED model is evaluated on the validation dataset for the DCASE 2024 Challenge Task 4, in terms of class-score-based polyphonic sound detection score (PSDS) and macro-average partial area under the receiver operating characteristic curve (MpAUC). The results show that the proposed model performs better than the baseline. In addition, the proposed SED model employing the multi-channel log-mel feature, auxiliary decoder, and MPA outperforms the baseline model. Ensembling several versions of the proposed SED model improves PSDS and MpAUC, scoring 0.038 higher in the sum of PSDS and MpAUC compared to the baseline model.

Index Terms— Sound event detection (SED), semi-supervised learning, auxiliary decoder, multi-channel log-mel feature, maximum probability aggregation

1. INTRODUCTION

Sound event detection (SED) aims to localize and classify individual sound events originating from acoustic signals, along with their corresponding timestamps. In recent years, the use of deep learning for SED has been widely researched [1]. While the performance of SED is satisfactory in some applications, such as [2, 3], a major challenge for developing deep learning-based SED models still remains in view of the preparation of label audio data with timestamps, which is expensive and time-consuming. This has prompted the development of weakly supervised and semi-supervised learning techniques [4] based on weakly labeled and unlabeled datasets [5]. Recently, a soft label–based dataset, called the Multi-Annotator Estimated STROng labels (MAESTRO) dataset [6], has also been employed to reduce the overall cost of annotating strong labels while maintaining the timestamps of sound events.

However, the use of mixtures of differently labeled data for SED yields a time misalignment problem that an inconsistency arises in the time recording units between the heterogeneously labeled datasets. In other words, soft labels contain label information over 1 s recording unit, whereas weakly labeled and unlabeled datasets, e.g., the Domestic Environment Sound Event Detection (DESED) dataset, contain sound events recorded over shorter units than 1 s. In addition to this time misalignment problem, there is another mismatch problem in the recording characteristics of sound events in the different datasets.

Thus, this paper proposes a maximum probability aggregation (MPA) approach for SED to address the time misalignment between the DESED and MAESTRO datasets. In addition, to accommodate time-frequency patterns according to different recording characteristics, a multi-channel log-mel feature is extracted to help the SED model capture sound events from two different datasets.

The proposed MPA and multi-channel log-mel feature are applied to an SED model, named a frequency dynamic convolution (FDY) [7]–large kernel attention (LKA) [8]-convolutional recurrent neural network (CRNN) model, which was developed for the DCASE 2023 Challenge Task 4A [9]. The FDY–LKA-CRNN

[*] This work was supported in part by Hanwha Vision Co. Ltd., the Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korean government (MSIT) (No.2022-0-00963), and the "Practical Research and Development support program supervised by the GTI" grant funded by the GIST in 2024.

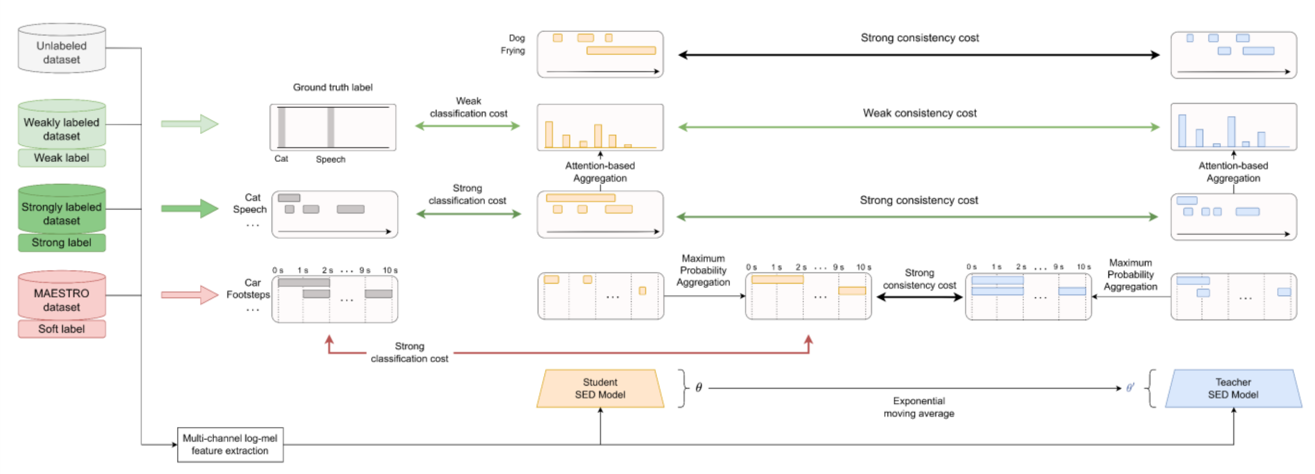


Figure 1. Illustration of the proposed SED model training procedure, focused on maximum probability aggregation.

model is trained via mean-teacher-based semi-supervised learning to handle unlabeled data, and it incorporates bidirectional encoder representation from audio transformer (BEATs) [10] embeddings to improve high-level semantic representation. However, the contribution of the BEATs encoder to the performance of the combined SED model is over-emphasized relative to that of the FDY–LKA-CRNN. To further improve the overall performance of the SED model, an auxiliary decoder [11] is applied to train the SED model with BEATs embeddings.

Our contributions can be summarized as follows:

- To deal with the time misalignment issue between the DESED and MAESTRO datasets, we propose MPA, which effectively aligns the time intervals between the predicted strong labels of the SED model and the soft labels in the MAESTRO dataset, thereby improving the overall performance of the SED model.
- To extract the heterogeneous time-frequency patterns of the sound events between the two datasets, we propose a multi-channel log-mel feature extraction method. Especially the feature improves a metric about MAESTRO dataset.
- Finally, we incorporate an auxiliary decoder to balance the contributions of the convolutional block and pretrained model by providing additional loss weighting during training. Consequently, the proposed auxiliary decoder-based training improves SED performance in both datasets.

The remainder of this paper is organized as follows: Section 2 describes the dataset and input features of the SED model developed in this study. Section 3 proposes a multi-channel log-mel feature and MPA, and also incorporates the auxiliary decoder for SED model training. Section 4 evaluates the performance of the developed SED model on the DCASE 2024 Task 4 validation dataset and compares the SED performance according to different combinations of the proposed approaches. Finally, Section 5 concludes this paper.

2. DATASET

Unlike in 2023, the database for the DCASE 2024 Challenge Task 4 comprises the DESED and MAESTRO datasets. The DESED dataset, which is identical to that for the last year's DCASE Challenge, contains several types of data such as weakly labeled data, unlabeled in-domain training data, strongly labeled synthetic data, and strongly labeled real data. All the audio clips span 10 seconds each. The weakly labeled dataset is composed of 1,578 clips with only class labels. The unlabeled in-domain training dataset contains 14,412 audio clips. Finally, the strongly labeled real and synthetic datasets contain 3,470 and 10,000 clips, respectively, where the strongly labeled synthetic dataset is created using Scraper [12]. Note that the number of audio event classes is 10 in this dataset.

The original MAESTRO dataset contains audio clips longer than 180 seconds. However, to balance the length of audio clips in this dataset with that in the DESED dataset, the audio clips are cropped to 10 s, allowing a 9 s overlap between consecutively cropped audio clips. Each cropped audio clip is softly labeled into 10 vectors, where each vector is assigned to every segment of 1 s with a dimension of 19 for representing 19 audio event classes. Notice that the event classes in the DESED dataset are different from those in the MAESTRO dataset, except for two classes, e.g., "Speech" in DESED and "People Talking" in MAESTRO, and "Dishes" in DESED and "Cutlery and dishes" in MAESTRO. After merging the similar two classes, there are 27 classes in total.

The mono-channel signals in the two datasets are first resampled from 44.1 to 16 kHz to extract audio features. Then, the audio signals are segmented into frames of 2,048 samples with a hop length of 160 samples. A 2,048-point fast Fourier transform is applied to each frame, followed by a 128-dimensional mel-filterbank analysis. Each 10 s audio clip comprises 1,001 frames. Hence, the input feature dimensions are 1001×128. The retrieved mel-spectrogram features are then normalized based on the mean and standard deviation for all training audio samples. When extracting the multi-channel log-mel feature, we use identical parameters for preprocessing.

3. PROPOSED METHOD

The SED model is based on the FDY–LKA-CRNN architecture that was proposed in [9], and it is trained via semi-supervised learning in a mean-teacher framework. Fig. 1 shows the proposed

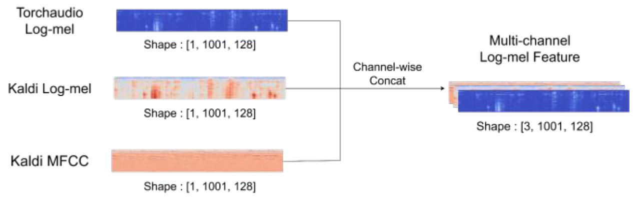


Figure 2. Illustration of the proposed multi-channel log-mel feature extraction procedure for obtaining the heterogeneous time-frequency patterns of the sound events.

SED model training procedure, where the newly proposed approaches, such as MPA and the multi-channel log-mel feature, are exaggerated. In addition to MPA and the multi-channel feature, the auxiliary decoder is intrinsically used for training the student and teacher models shown at the bottom of the figure. The following subsections sequentially describe MPA, the multi-channel feature, and the auxiliary decoder in detail.

3.1. Multi-channel log-mel feature

As mentioned in Section 2, there are different recording environments between the DESED and MAESTRO datasets, which are recorded in almost clean and noise conditions, respectively. To capture the diverse acoustic properties of the two datasets, we extract the multi-channel log-mel feature composed of 1) a log-mel spectrogram extracted using the Torchaudio framework, 2) a log-mel spectrogram extracted using Kaldi within the Torchaudio framework, and 3) the mel-frequency cepstral coefficient (MFCC) feature extracted using Kaldi within the Torchaudio framework.

Fig. 2 illustrates the proposed multi-channel log-mel feature extraction procedure for obtaining the heterogeneous time-frequency patterns of the sound events. First, three different feature vectors, as described above, are extracted and then concatenated channel-wise to create a multi-channel log-mel feature. This concatenated feature vector is input to the SED model during both training and inference. By leveraging multiple configurations to extract the log-mel features, it is expected that we create a robust input representation that effectively bridges the gap between the DESED and MAESTRO datasets.

3.2. Length-adjustable maximum probability aggregation

The FDY–LKA-CRNN-based SED model was developed for the DESED dataset, where audio data labels were assigned in segments less than 1 s. To accommodate different labels for sound events as in the MAESTRO dataset, we need to incorporate new techniques into the SED model. This is because the difference in labeling presents a significant challenge due to the mismatch in time intervals between the label information of the MAESTRO dataset and DESED dataset.

To deal with such a time misalignment problem, we propose the MPA. Compared to the labels in the DESED dataset, the soft labels in the MAESTRO dataset do not guarantee that a sound event entirely exists within each 1 s segment. The output of the SED model consists of predictions for 25 frames, which corresponds to a duration of 1 s. As shown in Figure 3, we select the highest probability value among these 25 frames and use this value as the class probability for the corresponding 1 s segment. This approach ensures that the time interval for the MAESTRO dataset would be aligned with the soft labels. This MPA is performed only during the training step.

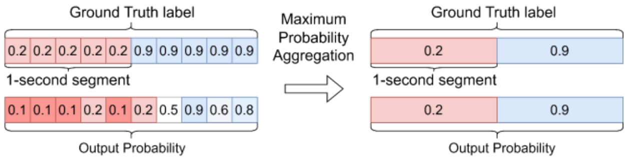


Figure 3. Example of implementing maximum probability aggregation, which is applied only to the MAESTRO dataset.

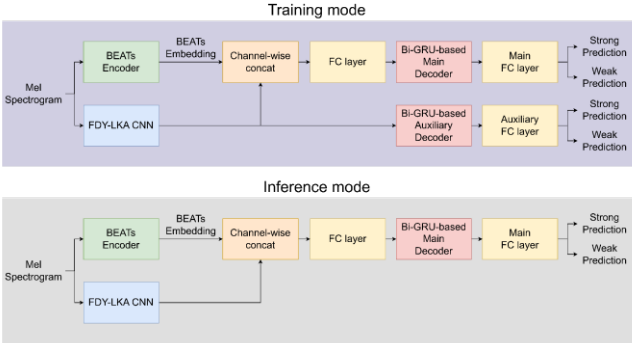


Figure 4. Network architecture of the proposed auxiliary decoder applied to train the FDY-LKA-CNN-based SED model with BEATs embeddings.

3.3. Auxiliary decoder

The BEATs encoder can extract the embedding corresponding to high-level semantic information, resulting in providing improved SED performance [9]. However, the contribution of the BEATs encoder to the performance of the combined SED model is overemphasized relative to that of the FDY–LKA-CRNN. Thus, we incorporate an auxiliary decoder to balance the contributions between the convolutional block and BEATs encoder by providing additional loss weighting during training.

Fig. 4 shows the network architecture of the proposed auxiliary decoder applied to train the FDY-LKA-CNN-based SED model with BEATs embeddings. The proposed auxiliary decoder mirrors the structure of the main decoder, consisting of two bidirectional gated recurrent units (Bi-GRUs) designed to capture temporal context information, followed by a fully connected (FC) classifier that uses a sigmoid function to calculate class probabilities. The auxiliary decoder does not share weights with the main decoder. Also, it is activated only during the training step, and a higher weight is assigned to the auxiliary loss in the initial training steps than the main loss. This guides the learning process so that the convolutional blocks are well-trained compared to without using the auxiliary decoder. During inference, the main decoder is only operated to generate the output of the SED model.

4. EXPERIMENTAL RESULTS

4.1. Model training

The parameters of the FDY–LKA-CRNN-based SED model were initialized through Xavier initialization [13]. The minibatch-wise adaptive moment estimation optimization technique [14] was employed, which involved decoupling the weight decay from the gradient-based updates. In addition, a dropout method [15] was applied to the FDY–LKA-CRNN model at a rate of 0.5. The learn-

Table 1: Performance comparison of the baseline and various versions of the proposed SED model on the validation dataset of DCASE 2024 Challenge Task 4.

Model	Auxiliary decoder	Maximum probability aggregation	Multi-channel log-mel feature	Ensemble	Validation Dataset		
					Class-score-based PSDS	MpAUC	Sum of metrics
Baseline: CRNN-based mean-teacher model [22]	–	–	–	–	0.49 ± 0.004	0.73 ± 0.007	1.22
FDY–LKA-CRNN	–	–	–	–	0.4799	0.665	1.144
FDY–LKA-CRNN–A	√	–	–	–	0.4922	0.673	1.164
FDY–LKA-CRNN–M	–	√	–	–	0.4959	0.692	1.187
FDY–LKA-CRNN–C	–	–	√	–	0.4663	0.709	1.175
FDY–LKA-CRNN–AM	√	√	–	–	0.5092	0.709	1.218
FDY–LKA-CRNN–MC	–	√	√	–	0.4832	0.733	1.216
FDY–LKA-CRNN–AC	√	–	√	–	0.4795	0.712	1.191
FDY–LKA-CRNN–AMC	√	√	√	–	0.5018	0.740	1.241
FDY–LKA-CRNN–AMC(E)	√	√	√	√	0.5162	0.742	1.258

ing rate was set based on the ramp-up strategy [4], with the maximum value reaching 0.001 after 50 epochs. Several augmentation techniques were applied to the train data, including time-frequency shift [16], time mask [17], mix-up [18], and filter augmentation [19].

4.2. Discussion

The performance of the proposed SED model was evaluated using the measures defined in the DCASE 2024 Challenge Task 4 [20]: class-score-based polyphonic sound detection score (PSDS) [21] and macro-average partial area under the receiver operating characteristic curve (MpAUC).

Table 1 compares the performance of the baseline with those of various versions of the proposed SED model on the validation dataset of the DCASE 2024 Challenge Task 4. As shown in the table, there are nine different versions in this study. The FDY–LKA-CRNN is the SED model identical to that in [9], which was developed in the DCASE 2023 Challenge. Then, we applied each of the three proposed approaches, such as auxiliary decoder, MPA, and multi-channel log-mel feature that are abbreviated as A, M, and C, respectively. For example, FDY–LKA-CRNN–A means the FDY–LKA-CRNN-based SED model trained using the proposed auxiliary decoder. The FDY–LKA-CRNN–AMC(E) means an ensemble model combined with the FDY–LKA-CRNN–AMCs obtained from 16 different checkpoints.

First of all, we observed the performance of FDY–LKA-CRNN SED model was degraded compared to that of the baseline model. This was because FDY–LKA-CRNN model was optimized to the labeling of the DESED dataset, as mentioned earlier. Then, we applied each of the three proposed approaches (A, M, and C) to FDY–LKA-CRNN. As shown from the third to fifth row in the table, any FDY–LKA-CRNN–X improved MpAUC compared to FDY–LKA-CRNN, while FDY–LKA-CRNN–C provided a little lower class-score-based PSDS than FDY–LKA-CRNN. However, combining any two out of three approaches achieved higher or comparable class-score-based PSDS and MpAUC to FDY–LKA-CRNN.

Next, we combined all the three approaches to construct FDY–LKA-CRNN–AMC. Then, it was revealed that FDY–LKA-CRNN–AMC yielded better than FDY–LKA-CRNN as well as the baseline model.

Finally, we constructed an ensemble model, FDY–LKA-CRNN–AMC(E), and compared its performance with the baseline and FDY–LKA-CRNN-based single models. As shown in the table, this ensemble model outperformed the baseline as well as the other single models. This superior performance was ascribed to the inherent advantages of ensemble modeling, such as reduced overfitting and improved model robustness.

5. CONCLUSIONS

In this paper, we proposed maximum probability aggregation and a multi-channel log-mel feature to improve SED performance when the training datasets were heterogeneously recorded and labeled. In addition, the auxiliary decoder-based training approach was proposed to balance the contributions of different representations prior to a classifier. In particular, our baseline model was FDY–LKA-CRNN with BEATs embeddings; thus, the auxiliary decoder could help the classifier get balanced information between the CNN block and the BEATs encoder. In summary, the auxiliary decoder enhanced the performance of the convolutional block, enabling it to extract semantics. MPA was applied to the MAESTRO dataset to match the time alignment between the output of the SED model and the soft labels. The multi-channel log-mel feature could help the SED model accommodate the various time-frequency patterns from the two different datasets used in this challenge. We constructed the SED model according to the rules of the DCASE 2024 Challenge Task 4. The experimental results showed that the SED model trained with the multi-channel log-mel feature, MPA, and auxiliary decoder increased the PSDS and MpAUC by 0.0118 and 0.01, respectively, compared to the baseline SED model. An ensemble model derived from the model checkpoints also improved the sum of PSDS and MpAUC by 0.038 over the baseline model.

In future work, we will investigate the effectiveness of the proposed approaches according to different neural architectures of SED models.

6. REFERENCES

[1] T. Khandelwal, R. K. Das, and E. S. Chng, “Sound event detection: A journey through DCASE Challenge series,” *APSIPA Trans. Signal Inf. Process.*, vol. 13, 2024.

[2] Y. R. Pandeya, B. Bhattarai, and J. Lee, “Visual object detector for cow sound event detection,” *IEEE Access*, vol. 8. pp. 162625–162633, 2020.

[3] S. Mohmmad, and S. K. Sanampudi. “Exploring current research trends in sound event detection: A systematic literature review,” *Multimedia Tools and Applications*, pp. 1–43, 2024.

[4] A. Tarvainen and H. Valpola, “Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results,” in *Proc. International Conference on Neural Information Processing Systems (NIPS)*, 2017, pp. 1195–1204.

[5] N. Turpault, R. Serizel, A. Shah, and J. Salamon, “Sound event detection in domestic environments with weakly labeled data and soundscape synthesis,” in *Proc. Workshop on Detection and Classification of Acoustic Scenes and Events*, 2019.

[6] I. Martín-Morató and A. Mesaros, “Strong labeling of sound events using crowdsourced weak labels and annotator competence estimation,” *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 31, pp. 902–914, 2023.

[7] H. Nam, S.-H. Kim, B.-Y. Ko, and Y.-H. Park, “Frequency dynamic convolution: Frequency-adaptive pattern recognition for sound event detection,” *arXiv preprint*, arXiv:2203.15296, 2022.

[8] M.-H. Guo, C.-Z. Lu, Z.-N. Liu, M.-M. Cheng, and S.-M. Hu, “Visual attention network,” *arXiv preprint*, arXiv:2202.09741, 2022.

[9] J. W. Kim, S. W. Son, Y. Song, H. K. Kim, I. H. Song, and J. E. Lim, “Label filtering-based self-learning for sound event detection using frequency dynamic convolution with large kernel attention,” in *Proc. Workshop on Detection and Classification of Acoustic Scenes and Events*, 2023.

[10] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, Z. Chen, and F. Wei, “BEATs: Audio pre-training with acoustic tokenizers,” *arXiv preprint*, arXiv:2212.09058, 2022.

[11] Y Jia, R. J. Weiss, F. Biadsy, W. Macherey, M. Johnson, Z. Chen, and Y. Wu, “Direct speech-to-speech translation with a sequence-to-sequence model,” *arXiv preprint,* arXiv:1904.06037, 2019.

[12] J. Salamon, D. MacConnell, M. Cartwright, P. Li, and J. P. Bello, “Scaper: A library for soundscape synthesis and augmentation,” in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2017, pp. 344–348.

[13] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proc. International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010, pp. 249–256.

[14] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint*, arXiv:1412.6980, 2014.

[15] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *J. Mach. Learn. Res.*, vol. 15, no. 56, pp. 1929–1958, 2014.

[16] L. Delphin-Poulat and C. Plapous, “Mean teacher with data augmentation for DCASE 2019 Task 4,” *Tech. Rep. in DCASE 2019 Challenge*, 2019.

[17] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, “Specaugment: A simple data augmentation method for automatic speech recognition,” *arXiv preprint*, arXiv:1904.08779, 2019.

[18] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “Mixup: Beyond empirical risk minimization,” *arXiv preprint*, arXiv:1710.09412, 2017.

[19] H. Nam, S.-H. Kim, and Y.-H. Park, “Filteraugment: An acoustic environmental data augmentation method,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 4308–4312.

[20] https://dcase.community/challenge2024/task-sound-event-detection-with-heterogeneous-training-dataset-and-potentially-missing-labels.

[21] J. Ebbers, R. Haeb-Umbach, and R. Serizel, “Threshold independent evaluation of sound event detection scores,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 1021–1025.

[22] https://github.com/DCASE-REPO/DESED_task/tree/master/recipes/dcase2024_task4_baseline.

LARGE-LANGUAGE-MODEL-BASED CAPTION AUGMENTATION FOR LANGUAGE-QUERIED AUDIO SOURCE SEPARATION

Yoonah Song[1*], *Do Hyun Lee*[1*], *and Hong Kook Kim*[1,2,3]

[1] AI Graduate School, [2] School of EECS GIST, Gwangju 61005, Republic of Korea
[3] Aunion AI, Co. Ltd., Gwangju 61005, Republic of Korea
{yyaass0531@gm., zerolee12@gm., hongkook@}gist.ac.kr

ABSTRACT

This paper proposes a prompt-engineering-based caption augmentation approach for enhancing the performance of language-queried audio source separation (LASS) models. In the context of LASS, when large language models (LLMs) are utilized to generate augmented captions for audio clip descriptions, the choice of LLM prompts significantly influences the performance of LASS models. Hence, this study compares the performance of a LASS model using a dataset-dependent prompt (DDP) and a dataset-independent prompt (DIP). Experimental results on a small-sized benchmarking dataset reveal that the DDP-based caption augmentation approach achieves better speech quality than the corresponding DIP approach. However, not all DDP-generated captions guarantee quality improvement of the LASS models. Thus, a criterion is proposed to exclusively select effective captions based on their Bidirectional Encoder Representations from Transformers (BERT) similarity scores relative to the original caption. Subsequently, augmented captions with BERT similarity scores exceeding a predefined threshold are adopted for model training. The effectiveness of the proposed prompt-engineering-based approach is then evaluated on the baseline LASS model of DCASE 2024 Challenge Task 9. Performance evaluation results show that the baseline LASS model using the proposed prompt-generated caption outperforms the model using the original caption. The proposed prompt-engineering approach is also applied to AudioSep, a state-of-the-art model, to verify its validity across diverse LASS models. Ablation studies reveal that selecting appropriate prompts for LLM-based caption augmentation significantly enhances LASS performance. Furthermore, selective augmentation based on BERT similarity scores can further boost audio separation quality.

Index Terms— Language-queried audio source separation (LASS), large language model (LLM), caption augmentation, BERT similarity score, DCASE 2024 Challenge Task 9

1. INTRODUCTION

Source separation refers to the technique of isolating specific sound sources from a mixture of audio signals. Traditionally, this domain has primarily focused on tasks with predefined target sources, including speech enhancement [1], speech separation [2], and music source separation [3]. Recently, significant research efforts have been devoted toward universal sound separation [4], which seeks to segregate diverse real-world sound classes. However, owing to the vast number of possible sound sources, predefining all potential classes is nearly impossible.

To address this, researchers have explored a query-based sound separation approach utilizing visual [5] or audio queries [6] to separate specific sound sources. One such approach is language-queried audio source separation (LASS) [7], [8], which leverages natural language queries to identify and separate target sound sources. However, significant challenges related to data availability and quality hinder the training of deep learning models for LASS. Furthermore, the effectiveness of the language-query approaches relies on the user of the LASS model. This implies that the manner in which an audio clip is queried can vary widely depending on the time, location, and occasions of using the LASS model, even when the same user is involved. Consequently, annotating individual audio clips with inputs from numerous people is essential [9]. However, this annotation process is expensive and time-consuming, resulting in a limited number of captions for each audio clip. To address this data scarcity, the most intuitive solution is to utilize text augmentation techniques.

Notably, text augmentation research in the natural language processing (NLP) domain aims to improve the robustness of NLP models by generating diverse yet meaningful variations of original sentences. Easy Data Augmentation [10], a representative example of text augmentation approaches, adopts four techniques: synonym replacement, random insertion, random swap, and random deletion. These techniques enhance the robustness of text classifiers through text augmentation. Beyond textual content, audio and video captioning tasks aim to convert non-textual media into descriptive language, thus enhancing the accessibility and comprehension of audiovisual content. While audio captioning tasks generate textual descriptions of sound content [11], video captioning tasks automatically generate textual descriptions of actions and events depicted in videos [12]. These tasks, similar to LASS, involve handling both textual information and audiovisual content. However, current captioning research, such as [11] and [12], predominantly focuses on augmenting audio and video features to address the challenges related to data scarcity. Consequently, attempts to augment a linguistic expressions remain scarce.

Unlike traditional multimodal data augmentation approaches, which focus on diversifying audio and video content, our research focuses on augmenting data to enrich textual diversity. Specifically, by augmenting captions, our approach enables the LASS models to identify and utilize diverse captions conveying the same meaning. For example, identifying "a sound of thin plastic rattling"

[*] Y. Song and D. H. Lee equally contributed.

[**] This work was supported in part by the GIST-MIT Research Collaboration grant and the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (RS-2024-00411137). It was also supported by "Project for Science and Technology Opens the Future of the Region" program through the Innopolis Foundation funded by Ministry of Science and ICT (Project Number: 2022-DD-UP-0312).

Table 1: Summary of the training datasets used in this paper.

Dataset	Data subset	# of Clips	# of Captions
FSD50K		40,966	40,966
Clotho v2		3,839	19,195
WavCaps	BBC	31,201	31,201
	SoundBible	1,232	1,232
	AudioSet	108,317	108,317

as "fire crackling." Additionally, utilizing multiple captions for each audio clip enhances LASS performance. To further enhance the performance of LASS models, this study utilizes a large language model (LLM) to augment the caption of audio clips. Recently, numerous studies have developed approaches for text augmentation using LLMs, highlighting the significant impacts of LLM input prompts on the output quality [13]. Considering this, the current study investigates sophisticated prompt designs to enhance LLM-based caption augmentation. The key contributions of our research are summarized as follows:

- We first investigate how to effectively design an input prompt for augmenting captions using an LLM. Our results indicate that a dataset-dependent prompt (DDP), which is designed considering various sentence structures across different datasets, performs better than a dataset-independent prompt (DIP), which uses a single prompt regardless of the dataset.
- Given that all generated captions by LLM may not necessarily improve the training of LASS models, we establish a criterion for selecting captions. This criterion adopts the BERT similarity score to quantify the similarity between original and augmented captions. Subsequently, performance evaluations of the LASS model are conducted by selecting captions depending on their similarity scores. Our findings show that utilizing descriptive captions with a diverse range of similarity scores is more effective than focusing solely on those with high similarity to the original ones.
- We examine the effectiveness of the proposed prompts and selection criterion across different LASS models. The results reveal that the proposed approach demonstrates effective for the baseline model of DCASE 2024 Challenge Task 4 and the AudioSep model [8]. Consequently, the LASS model employing our caption augmentation approach is ranked first in the evaluation of DCASE 2024 Challenge Task 9.

The remainder of this paper is organized as follows. Section 2 describes the datasets used for the LASS model. Section 3 presents the LASS model and proposes our caption augmentation approach using an LLM-based prompt-engineering strategy. Section 4 presents a performance evaluation of the LASS model on the validation dataset of DCASE 2024 Challenge Task 9. Finally, Section 5 concludes this paper.

2. DATASET

The baseline system of DCASE 2024 Challenge Task 9 [7], [8] was developed using audio samples sourced from Clotho v2 [9] and Freesound Dataset 50K (FSD50K) [14]. Notably, all audio clips within these datasets were acquired from the Freesound platform. In FSD50K, captions for each audio clip were initially obtained by refining raw descriptions using ChatGPT. Meanwhile, captions in Clotho v2 were crowdsourced using annotators from English-speaking countries, resulting in five captions per audio clip. In addition to these datasets, WavCaps [15] dataset was also used as one of externally available datasets. The WavCaps [15] dataset included the Freesound subset, but the Freesound subset was excluded in this work. More detailed information regarding the selected datasets is outlined in Table 1.

In particular, FSD50K was an extensive dataset comprising 51,197 Freesound clips with human-labeled sound occurrences. Each audio clip was categorized based on 200 AudioSet ontology classes. Clotho v2 was an audio captioning dataset comprising 5,929 audio clips. Of these, 3,839 audio clips are allocated for development, 1,045 for validation, and 1,045 for evaluation. Each clip possessed five manually generated captions, varying in length from eight to twenty words. Finally, WavCaps in this study comprised 140,750 audio clips excluding the Freesound subset. Meanwhile, captions for these audio clips were generated by ChatGPT based on their raw audio descriptions. While consistent description-generation conditions were applied to SoundBible, and BBC, differing conditions were adopted for AudioSet.

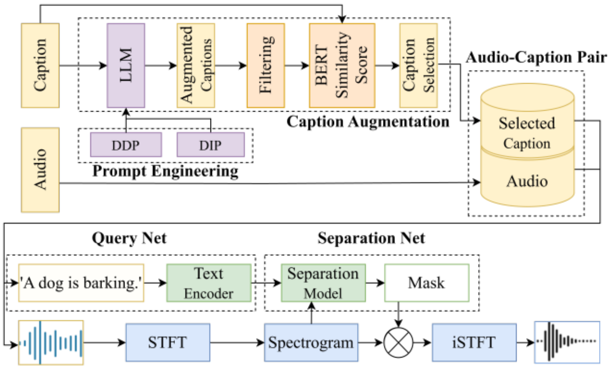


Figure 1: Procedure of training a LASS model using the proposed caption augmentation approach.

Table 2: Distribution of captions according to different datasets and prompts.

# of Captions / Dataset		Original	DDP	DIP
FSD50K		162,485	40,966	30,991
WavCaps	BBC	31,201	131,969	27,225
	SoundBible	1,232	1,997	550
	AudioSet	108,319	406,139	75,300

3. PROPOSED LASS MODEL

We develop a LASS model based on the baseline model of DCASE 2024 Challenge Task 9 [7], [8]. This baseline LASS model comprises two key components: Query Net, which leverages Contrastive Language Audio Pretraining (CLAP) [16], and Separation Net which employs ResUNet [17]. Notably, the desired target source is conditioned by Query Net and separated by Separation Net. This paper proposes a prompt-engineering-based approach for LLM based caption augmentation, aiming to diversify the query representations of Query Net as if annotated by many humans.

Fig. 1 illustrates the training procedure of our LASS model using the proposed caption augmentation approach. Initially, a prompt corresponding to original caption of an audio clip is inputted into an LLM with prompt to generate multiple captions. The resulting captions are subsequently filtered to remove a prompt with the original sentences. The filtered captions are then incomplete or interrogative caption corresponding to an audio clip, compared to the original caption, in terms of their meaningfulness and

Table 3: Comparison of SDR, SDRi, and SI-SDR between DDP and DIP with randomly sampled 10,000 audio clips from different datasets.

Dataset	FSD50K				WavCaps							
					BBC+SoundBible				AudioSet			
Prompt Type	Original	DDP	DDP	DIP	Original	DDP	DDP	DIP	Original	DDP	DDP	DIP
# of Augmented Captions	-	39,586	7,498	7,498	-	41,359	8,578	8,578	-	37,634	6,879	6,879
SDR	3.960	4.432	4.237	4.113	4.577	4.968	4.807	4.764	4.465	5.038	4.808	4.764
SDRi	3.925	4.397	4.202	4.078	4.542	4.933	4.772	4.729	4.430	5.003	4.773	4.729
SI-SDR	0.293	1.470	0.852	0.618	1.083	2.431	2.090	1.961	1.092	2.311	2.215	1.848

Table 4: Comparison of SDR, SDRi, and SI-SDR in relation to BERT similarity scores for FSD50K dataset, using approximately 11,277 randomly selected augmented captions.

Threshold	0	0.700	0.850
# of Augmented Captions	11,277	11,277	11,277
SDR	4.236	4.369	4.167
SDRi	4.201	4.334	4.132
SI-SDR	1.195	1.372	1.080

diversity, using BERT-based similarity scores [18]. Next, the selected captions are paired with their corresponding audio clips, forming multiple pairs of caption-audio clips by copying the original audio cli p to make the pairs. Finally, the LASS shown in the lower arm of the figure is trained using the augmented pairs of caption-audio clips. Note here that Microsoft's Phi-2.0 LLM [19] is used for caption generation, which is a 2.7 billion-parameter language model known for its superior comprehension and generation capabilities compared to the Llama-7B model.

3.1. Quality of augmented captions based on input prompts

We first investigated how to effectively design an input prompt for augmenting captions using the LLM. A review of relevant studies revealed that individual datasets require unique prompts customized to their specific attributes rather than general prompts [16]. Therefore, we hypothesized that crafting prompts tailored to the attributes of each dataset could enhance the quality of the generated captions. This approach led to the development of dataset-dependent prompts (DDPs), which generate sentences closely resembling original descriptions while meeting prompt requirements. On the other hand, the dataset-independent prompts (DIPs) were designed to be applicable even without prior information of individual dataset characteristics.

Second, we customized DDP based on the caption-generation conditions adopted in [9], [14], [15] curating a distinct prompt for each of the FSD50K, AudioSet subset, and BBC+SoundBible subset. Here, SoundBible+BBC means a subset datasets combining the BBC and SoundBible subsets because they share identical caption-generation conditions [15].

Next, to formulate a DIP, we initially referenced the caption-generation prompts utilized in WavCaps [15] and Clotho v2 [9]. However, they used a prompt to generate a sentence by only considering the event label. It was evident that these prompts might not be suitable for our study because we needed to augment sound description captions at the sentence level but not the event word level. To remedy this issue, we needed to redesign the DIP so that it could consider a sentence with a similar meaning to the original prompt. The detailed d esign process of the DIP is described in [20]. After generating the captions using the DDP or DIP, the captions were filtered and selected, as described above. In particular, the BERT similarity score between the original and each generated caption was computed. Then, captions whose scores were higher than a predefined threshold were selected, while captions with a similarity score of 1.0 was removed because it implied that the generated ca0ptions were identical to the original caption. Table 2 presents the number of captions augmented using each prompt for different datasets after applying the filtering process to the generated captions.

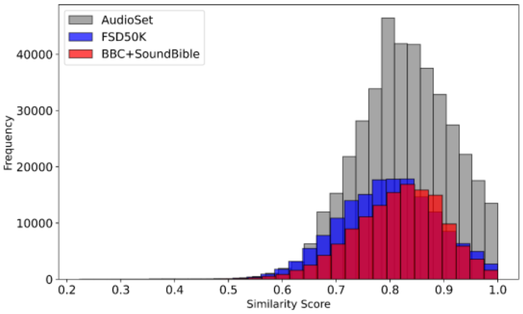


Figure 2: Distribution of BERT similarity scores for DDP-based augmented captions.

Table 3 compares the signal-to-distortion ratio (SDR) between the DDP and DIP. Due to the large data sizes, we randomly sampled 10, 000 audio clips from each dataset and used the corresponding caption-audio pairs for this experiment. While both types of prompts were designed to augment four captions per audio clip, the DIP produced approximately 1 to 1.5 captions on average, whereas the DDP consistently generated 4 captions on average. Specifically, the LASS model employing the DDP for FSD50K achieved an SDR of 4.432 dB, whereas that employing the DIP attained an SDR of 4.113 dB. Meanwhile, for the BBC+ SoundBible subset, the LASS models using the DDP and DIP achieved SDRs of 4.968 and 4.764 dB, respectively. Finally, the LASS models using the DDP and DIP achieved SDRs of 5.038 and 4.764 dB, respectively, for the AudioSet subset. As shown in the table, there was a similar tendency in other quality measures such as SDR improvement (SDRi) and scale-invariant (SI)-SDR, when comparing the DDP-based augmentation with DIP-based one. To ensure a fair quality comparison, we conducted additional experiments with equal numbers of augmented captions for both methods. Consequently, DDP consistently outperformed DIP in SDR and SI-SDR, indicating superior caption quality. It was also revealed that better performance was achieved with more captions for DDP.

Based on these results, it was proven that the DDP was more effective than the DIP, the LASS model employing DIP-based augmentation demonstrated performance improvement over that using original captions. Hence, in cases with limited knowledge regarding specific data characteristics, our DIP can be also useful as a viable alternative.

3.2. Quality of augmented captions based on the BERT similarity score

Since it is uncertain whether all captions automatically generated

Table 5: Performance comparison of different LASS models trained using various combinations of training datasets with/without caption augmentation on the validation dataset of DCASE 2024 Challenge Task 9.

Model	Training Dataset	Training Approach	Caption Augmentation	SDR	SDRi	SI-SDR
Baseline	Baseline Dev Set (FSD50K + Clotho v2)	Full	N/A	5.817	5.782	3.837
		Full	DIP	6.547	6.512	4.636
		Full	DDP	6.716	6.681	4.729
	Baseline Dev Set + WavCaps	Full	N/A	7.500	7.465	5.795
		Full	DIP	7.750	7.715	6.161
		Full	DDP	7.818	7.783	6.321
AudioSep	-	Pretrained	-	8.195	8.160	6.708
	Baseline Dev Set + WavCaps	Fine-tuning	N/A	8.370	8.335	7.109
		Fine-tuning	DIP	8.459	8.424	7.072
		Fine-tuning	DDP	8.489	8.454	7.198

by the LLM effectively contribute to the training of the LASS model, we establish a criterion for selecting captions. To this end, the BERT similarity score is used to measure the similarity between the original and augmented captions, because the BERT similarity score can assess the similarity of each token in the candidate sentence using contextual embeddings [18].

Fig. 2 depicts the distribution of BERT similarity scores for DDP-based augmented captions. Each distribution seems to be a Gaussian distribution with a mean of 0.85 and a little different variance. It was observed from the comparison between the original and generated captions that the generated captions with similarity scores below a certain threshold could be unsuitable as sound descriptions. For instance, the original caption “A musician plays a tune on a wind instrument” was augmented to “The sound of thunder fills the air, shaking the ground and captivating everyone's attention,” scoring 0.6, thus significantly differing in meaning.

Next, a performance evaluation of the LASS model on the FSD50K dataset was conducted by selecting captions depending on the BERT similarity score. Table 4 compares the objective performance of the LASS models trained by the selected captions according to different thresholds, where approximately 11,277 randomly selected captions were used for each threshold. As shown in the table, we set the threshold as 0.7 for caption selection, because using descriptive captions with a diverse range of similarity scores was more effective than using those with high similarity to the original ones.

4. PERFORMANCE EVALUATION

In this section, we evaluated the performance of the LASS models employing DDP and DIP. In addition to the baseline LASS model, the AudioSep model [8] was also trained to examine the effectiveness of the proposed prompts and selection criterion on different LASS models. Table 5 compares SDR, SDRi, and SI-SDR of different LASS models trained using various combinations of training datasets with/without caption augmentation on the validation dataset of DCASE 2024 Challenge Task 9. In this work, the Adam optimizer with a learning rate of 1×10^{-3} and a batch size of 64 was applied for 100 epochs to train the LASS models. Notice that the BERT similarity score threshold of selecting captions was all set to 0.7.

As shown in Table 5, the baseline LASS model trained on Baseline Dev Set achieved an SDR of 5.817 dB, which is consistent with the DCASE 2024 Challenge Task 9 baseline checkpoint [21], [22]. Augmenting the Baseline Dev Set with DIP-based generated captions increased the SDR to 6.547 dB, and DDP-based captions further improved it to 6.716 dB, demonstrating better performance compared to DIP-based ones. Training on the WavCaps dataset (excluding Freesound) resulted in an SDR of 7.500 dB, with DDP-based captions pushing the SDR to 7.818 dB.

Next, the pretrained AudioSep model, which was trained on over 2 million clips from weakly labeled datasets such as AudioSet, VGGSound, and AudioCaps, was utilized to validate the general applicability of the DDP-based caption generation approach. As shown in Table 5, the pretrained AudioSep model achieved an SDR of 8.195 dB, surpassing that of the baseline LASS model trained with DDP-based augmented captions. Fine-tuning this model using Baseline Dev Set and WavCaps dataset increased the SDR to 8.370 dB. On the other hand, the AudioSep model using DDP-based generated captions reached SDR of 8.489 dB, demonstrating performance compared to that using DIP-based captions. Thus, the AudioSep model fine- tuned by employing DDP-based caption augmentation demonstrated the best performance in terms of the SDR, SDRi, and SI-SDR.

5. CONCLUSION

To enhance the performance of LASS models, this paper proposed DDP-based caption augmentation as a means of prompt engineering. Specifically, two prompts were developed: a DDP, which is tailored to the characteristics of a specific dataset, and a DIP, which could be used without dataset information. Utilizing these prompts, five captions were generated for each audio clip using an LLM, following which selective learning of the augmented captions was performed based on BERT similarity scores. Subsequently, the SDR performance of the baseline and AudioSep models with DDP-based and DIP-based caption augmentation was assessed. Our findings demonstrated that the DDP, which dependently considered the unique characteristics of each dataset, yielded more suitable results. Furthermore, performance improvements were observed as the BERT similarity scores between the original and augmented captions reached values of 0.700 or higher. Collectively, these findings underscore the importance of customized prompt engineering in enhancing LASS performance through data augmentation.

In our study, we utilized LLM to augment captions to enhance the performance of the LASS model. While our approach showed improved results, several limitations should be noted. Primarily, the use of LLMs and the design of appropriate prompts for caption augmentation are still largely unexplored. Thus, our approach may not have fully leveraged the potential of the model. Additionally, although performance improvements were observed in LASS, it is uncertain if these enhancements can be generalized to other tasks using caption-audio paired data, such as audio captioning.

6. REFERENCES

[1] S. Pascual, A. Bonafonte, and J. Serra, "SEGAN: Speech enhancement generative adversarial network," in *Proc. Interspeech*, 2017, pp. 3642–3646.

[2] D. Wang and J. Chen, "Supervised speech separation based on deep learning: An overview," *IEEE/ACM Trans. Audio Speech Lang.Process.*, vol. 26, no. 10, pp. 1702–1726, 2018.

[3] S. Park, T. Kim, K. Lee, and N. Kwak, "Music source separation using stacked hourglass networks," *arXiv preprint* arXiv:1805.08559, 2018.

[4] I. Kavalerov, S. Wisdom, H. Erdogan, B. Patton, K. Wilson, J. Le Roux, and J. R. Hershey, "Universal sound separation," in *Proc. WASPAA*, 2019, pp. 175–179.

[5] R. Gao and K. Grauman, "VisualVoice: Audio-visual speech separation with cross-modal consistency," in *Proc. CVPR*, 2021, pp. 15490–15500.

[6] B. Gfeller, D. Roblek, and M. Tagliasacchi, "One-shot conditional audio filtering of arbitrary sounds," in *Proc. ICASSP*, 2021, pp. 501–505.

[7] X. Liu, H. Liu, Q. Kong, X. Mei, J. Zhao, Q. Huang, M. D. Plumbley, and W. Wang, "Separate what you describe: Language-queried audio source separation," in *Proc. Interspeech*, 2022, pp.1801–1805.

[8] X. Liu, Q. Kong, Y. Zhao, H. Liu, Y. Yuan, Y. Liu, R. Xia, Y. Wang, M. D. Plumbley, and W. Wang, "Separate anything you describe," *arXiv preprint,* arXiv:2308.05037, 2023.

[9] K. Drossos, S. Lipping, and T. Virtanen, "Clotho: An audio captioning dataset," in *Proc. ICASSP*, 2020, pp. 736–740.

[10] J. Wei and K. Zou, "EDA: Easy data augmentation techniques for boosting performance on text classification tasks," in *Proc. EMNLP-IJCNLP*, 2019, pp. 6382–6388.

[11] Z. Ye, Y. Wang, H. Wang, D. Yang and Y. Zou, "FeatureCut: An adaptive data augmentation for automated audio captioning," in *Proc. APSIPA*, 2022, pp. 313–318

[12] C. Wang, H. Yang, and C. Meinel, "Image captioning with deep bidirectional LSTMs and multi-task learning," *ACM TOMM*, vol. 14, issue 2s, pp. 1–20, 2018.

[13] Y. Zhou, A. Muresanu, Z. Han, K. Paster, S. Pitis, H. Chan, and J. Ba, "Large language models are human-level prompt engineers," in *Proc. ICLR*, 2023. Available: https://openreview.net/forum?id=92gvk82DE-.

[14] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "FSD50K: An open dataset of human-labeled sound events," *IEEE/ACM Trans. Audio Speech Lang. Process,* vol. 30, pp. 829–852, 2022.

[15] X. Mei, C. Meng, H. Liu, Q. Kong, T. Ko, C. Zhao, M. D. Plumbley, Y. Zou, and W. Wang, "WavCaps: A ChatGPT assisted weakly-labelled audio captioning dataset for audio language multimodal research," *arXiv preprint*, arXiv: 2303.17395, 2023.

[16] C. Li, M. Zhang, Q. Mei, W. Kong, and M. Bendersky, "Learning to rewrite prompts for personalized text generation," in *Proc. ACM Web Conference*, 2024, pp. 3367–3378.

[17] Q. Kong, Y. Cao, H. Liu, K.Choi, and Y. Wang, "Decoupling magnitude and phase estimation with deep ResUNet for music source separation," in *Proc. ISMIR*, 2021. Available: https://doi.org/10.48550/arXiv.2109.05418.

[18] T. Zhang, V. Kishore, F. Wu, K. Weinberger, and Y. Artzi, "BERTScore: Evaluating text generation with BERT," in *Proc. ICLR*, 2020. Available: https://openreview.net/forum?id=SkeHuCVFDr.

[19] M. Javaheripi and S. Bubeck, "Phi-2: The surprising power of small language models." Available at https://www.microsoft.com/en-us/research/blog/phi-2-the-surprising-power-of-small-language-models/, and accessed on May 01, 2024.

[20] D. H. Lee, Y. Song, and H. K. Kim, "performance improvement of language-queried audio source separation based on caption augmentation from large language models for DCASE Challenge 2024 Task 9," *arXiv preprint*, arXiv:2406.11248, 2024.

[21] L. Xubo and Z. Yan, "DCASE 2024 Task 9: Language-queried audio source separation | pre-trained weights for the baseline system." Available at https://zenodo.org/records/10887460, and accessed on May 01, 2024.

[22] https://dcase.community/challenge2024/task-language-queried-audio-source-separation. Accessed on May 01, 2024.

SALT: STANDARDIZED AUDIO EVENT LABEL TAXONOMY

Paraskevas Stamatiadis, Michel Olvera, Slim Essid

LTCI, Télécom Paris, Institut Polytechnique de Paris, France
{paraskevas.stamatiadis, olvera, slim.essid}@telecom-paris.fr

ABSTRACT

Machine listening systems often rely on fixed taxonomies to organize and label audio data, key for training and evaluating deep neural networks (DNNs) and other supervised algorithms. However, such taxonomies face significant constraints: they are composed of application-dependent predefined categories, which hinders the integration of new or varied sounds, and exhibits limited cross-dataset compatibility due to inconsistent labeling standards. To overcome these limitations, we introduce *SALT: Standardized Audio event Label Taxonomy*. Building upon the hierarchical structure of AudioSet's ontology, our taxonomy extends and standardizes labels across 24 publicly available environmental sound datasets, allowing the mapping of class labels from diverse datasets to a unified system. Our proposal comes with a new Python package designed for navigating and utilizing this taxonomy, easing cross-dataset label searching and hierarchical exploration. Notably, our package allows effortless data aggregation from diverse sources, hence easy experimentation with combined datasets.

Index Terms— Machine listening, DCASE, sound taxonomy, sound categorization, data aggregation

1. INTRODUCTION

Machine listening systems support a wide range of audio applications, including urban sound analysis [1, 2], industrial acoustic monitoring [3, 4, 5], music analysis [6, 7, 8], and speech recognition [9, 10, 11]. The key success of these systems, typically relying on supervised machine learning approaches, especially using deep neural networks (DNNs), lies in the systematic annotation of training data, using predefined class labels from hierarchical sound ontologies and taxonomies [12, 13, 14, 15, 16, 17, 18].

Sound ontologies and taxonomies serve as foundational frameworks to categorize everyday sound scenes and events [19]. Developed across several research fields—for instance auditory cognition [20, 21], soundscape research [22], sound design [23]—they are particularly instrumental in machine listening. Notable examples stand out for their wide adoption in the DCASE[1] community: UrbanSound8K [12], SONYC-UST [15, 24], MAVD-traffic [16] for urban sound analysis and ESC-50 [25] and AudioSet [15] for broader sound event recognition.

As evident from the previous examples, categorization of sounds in these systems are context-specific and tailored to desired applications. Their static nature often entails significant, overhauls for updates or extensions, especially when combining audio events from different environments, even for the same application. An exemplary case of this, is the adaptation of the SONYC-UST's taxonomy. Originally developed to classify urban sounds in New York City, this taxonomy required an expansion to accommodate the unique sounds of Singapore city's soundscapes, while maintaining compatibility with the base categorization. This adaptation allowed for bench-marking of urban sound tagging systems across different cities [17, 26].

While recent initiatives such as *mirdata* [27] and *Soundata* [28] have simplified the use of major datasets for Music Information Research (MIR) and DCASE, by standardizing data loading, these efforts primarily focus on addressing issues related to data management and accessibility through open-source software packages. As such, these packages promote reproducibility and flexible data-processing pipelines. However, the development of adaptable and extensive sound categorization frameworks capable of integrating new audio event labels from diverse datasets while maintaining compatibility with existing taxonomies remains largely unaddressed.

In this work, we tackle such challenges by introducing *SALT: a Standardized Audio event Label Taxonomy*. Leveraging the hierarchical structure of AudioSet, *SALT* extends and standardizes labels across 24 publicly available environmental sound datasets. Such a large collection of datasets covers diverse audio analysis tasks including audio tagging, sound event detection and acoustic scene classification. By standardizing labels, SALT enables mapping them across diverse datasets, ensuring compatibility and easing dataset aggregation. Alongside our proposed taxonomy of standard dataset labels, we present *py-salt*, an open-source python package designed to navigate through its content. This tool allows users to easily navigate through the hierarchical label taxonomy at any level of granularity. It turns out to be quite valuable when performing experiments considering various existing datasets whose particular labelling schemes can be seamlessly represented in our unified taxonomy.

We posit that our contribution is timely in a research context where large-scale training of audio models is fueled by the availability of (labelled) training data and computational resources. Our taxonomy with standardized audio event labels simplifies data aggregation, complementing tools like *Soundata* to develop audio classification models at scale.

The remainder of this work is organized as follows: Section 2, introduces the motivation and design principles behind SALT. Section 3 presents the functionalities and applications of py-salt, our proposed Python package, and Section 4 concludes the article.

2. SALT

The motivation behind creating SALT is the development of a new solution leveraging existing taxonomies to facilitate experimentation across different environmental sound datasets. The key feature of this solution is label aggregation, which allows unified categorization of sound events. This approach necessitates a standardized

[1] Detection and Classification of Acoustic Scenes and Events

set of labels applicable to multiple environmental sound datasets. Consequently, with SALT we aim to expand AudioSet, the largest general-purpose sound event taxonomy, and use it as a common frame of reference to represent the annotations of all major publicly available DCASE datasets.

2.1. Design Principles

We aim to establish a general-purpose sound taxonomy with label aggregation capabilities at the core of its design. To achieve this, we adapt existing sound event taxonomies from diverse domains, including but not limited to urban sound analysis, acoustic scene classification, domestic sound event detection, among others, using AudioSet's taxonomy as our basis. A key principle is to integrate labels from diverse sound collections, prioritizing datasets that are independent from each other rather than subsets of others, leading to a natural expansion of AudioSet's taxonomy. This integration into a unified taxonomy entails a standardization process to ensure label consistency across datasets.

Label standardization. The standardization process involves a mapping of original (*i.e.*, default) category names from different datasets that describe the same acoustic event to a standardized label. For example, labels such as "car horn" in *UrbanSound8K*, "car_horn" in *ESC-50* and "Vehicle horn, car horn, honking" in *AudioSet*, all refer to the sound produced by a car horn. To aggregate labels effectively, we map them to the standard label *car_horn* in SALT. Our notation for denoting standard labels uses lowercase characters and underscores instead of white spaces.

Mapping for accurate aggregation. In cases where a dataset label indicates more than one acoustic event, or sources producing sound, the mapping depends on the nature of the sounds. If the events or sources have similar acoustic properties, the word "or" is introduced in the standard label to preserve both sounds in the label. For example, the label "Railroad car, train wagon" in AudioSet is mapped to the standard label *railroad_car_or_train_wagon* as both sources produce the same type of sound. On the contrary, when a dataset label indicates multiple sound events, each of them entailing unique acoustic signatures, the mapping selects the most specific (*i.e.,*finest-grained) standard label that avoids incorrect associations in the aggregation process. For example, the label "dog-barking-whining" in *SONYC-UST* is mapped to the broader standard label *dog* to ensure accurate aggregation. This principle prevents mistakenly including unrelated events into more specific standard labels such as *dog_barking* or *dog_whining*. In Figure 1 we present a clear depiction of our label standardization procedure.

Hierarchy expansion. Additionally, our objective is to preserve the base hierarchy of AudioSet while integrating new standard labels when strictly necessary. This design principle serves two main purposes. First, it facilitates label aggregation across multiple hierarchical levels by mapping dataset labels not only to a standard label, but also to its hierarchical ancestors (also standardized labels). For example, the label "Bird" in AudioSet is mapped to the standard label *bird* in our taxonomy, as well as to its standard ancestors *wild_animal* and *animal*. Second, it refines the AudioSet taxonomy by incorporating new or rare sound event labels coming from a wide variety of environmental sound datasets serving different audio analysis tasks. When a dataset contains class labels which do not fit neatly into the AudioSet taxonomy or cannot be covered by any existing node in the structure, new standardized labels are introduced to accomodate such labels. For instance, labels such as "truck/compressor" from the *MAVD-traffic* and "Friction brake" from the *SINGA:PURA* datasets, represent cases where AudioSet's existing labels are insufficient to fully capture the diverse set of sounds encountered in publicly available datasets.

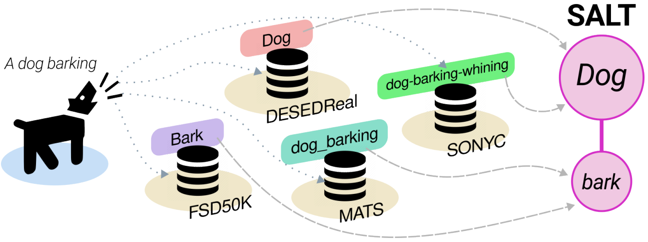


Figure 1: Illustration of SALT's standardization process. Dataset labels are systematically mapped to a standard label that ensures cross-dataset compatibility.

2.2. Taxonomy Structure

Our proposed extension to Audioset's taxonomy, is structured into multiple hierarchical levels, each representing a different granularity of sound categories. Starting from AudioSet's seven broad sound categories — "Human sounds", "Animal", "Music", "Source-ambiguous sounds", "Sounds of things", "Natural sounds", and "Channel, environment and background" — and 616 sound labels (out of the 632 provided in the taxonomy), we expand to 734 sound labels. These labels are categorized under the original seven AudioSet categories, with the addition of two new categories: *Water* and *Other*. Figure 2 illustrates the contribution of the original labels from all considered datasets to compose the standard labels in SALT.

With careful examination of the video clips available in AudioSet and their associated labels, we refined the hierarchical structure of AudioSet to clarify the placement of labels within the taxonomy. For example, when examining the label "Water", we found that 37% of clips include tags related to water sounds occurring in domestic environments *e.g., "Water faucet"*, while, only 2% of them are related to outdoor and/or natural landscapes. This distribution indicates that the "Water" tag does not exclusively belong under "Natural sounds", but also frequently appears in domestic settings. Additionally, we conducted refinements by examining children within categories such as "Vehicle" and "Engine". For example, clips tagged with the label "Accelerating, revving, vroom", categorized under "Engine", primarily pertain to vehicle sounds, accounting for approximately 93% of its instances. Therefore, "Accelerating, revving, vroom" is additionally categorized under "Vehicle". For a complete list of all such refinements, we refer the reader to our companion repository[2].

3. PY-SALT

In this section, we give an overview of the functionalities and applications of SALT, designed to unify event labels through standard

[2] https://github.com/tpt-adasp/salt

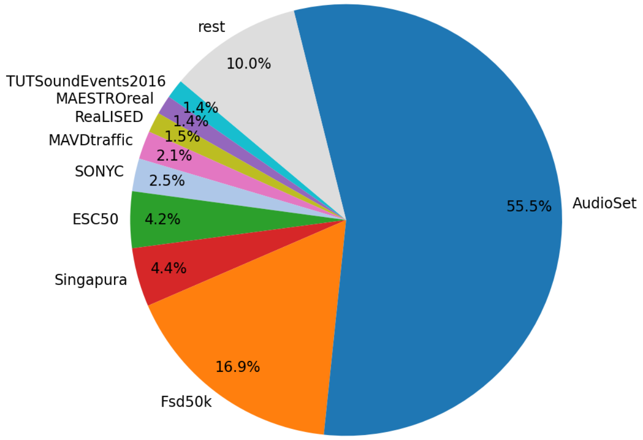


Figure 2: Contribution of dataset's original (default) labels to SALT after the standardization process.

labels, allowing for label search, data exploration, and hierarchical parsing. To exploit the benefits of our proposed taxonomy with standard labels, we developed *py-salt*, a Python package that provides tools for navigating and utilizing the taxonomy.

3.1. Functionalities

Label searching. This functionality allows two searching modes. First, standard labels can be employed to look for corresponding original (default) dataset labels across all those integrated in SALT, *e.g., motorcycle* → "motorbike" coming from *Urbansas*, "Motorcycle" from *AudioSet/FSD50K*, "motorcycle/wheel_rolling", "motorcycle/engine_idling", "motorcycle/engine_accelerating" from *MAVD-traffic*, etc. Secondly, original dataset labels can be employed to identify their counterparts across all datasets within the taxonomy. This dual approach offers comprehensive coverage and consistency for cross-dataset retrieval, *e.g., ReaLISED's* "water tap" → "Water_tap_and_faucet", "Water tap, faucet", "water tap running" coming from *FSD50K*, *AudioSet* and *TUT Sound Events 2016*, respectively.

Hierarchical exploration and expansion. This functionality allows browsing the taxonomy at any level of the hierarchy and easily locate superodinate (parent), subordinate (child) and coordinate (sibling) categories. Additionally, SALT supports *mapping expansion*, a functionality useful to incorporate new datasets and label categories into the taxonomy. The mapping process can be performed using the existing standard labels or by defining new ones suiting the user's requirements.

Visualization and searching tools. Graph plotting utilities are included in `py-salt`, which allows users to explore SALT visually. The python library contains methods to plot graphs showing the hierarchical structure of a given standard label in SALT, and also to depict all original (default) class names in the aggregated datasets that mapped to a SALT label. For example, the function `plot_hierarchical_tree_graph('bird')` serves to generate a graphical representation of the hierarchical structure for the standard label *bird* as illustrated in Figure 3. This functionality provides a clearer depiction of the relationships between parent, child and sibling categories. Similarly, the function `plot_std_label_mapping('car_horn')` serves to show the mapping of dataset labels to the standard label *car_horn* as illustrated in Figure 4. This is useful for identifying the potential datasets needed for a specific application of interest.

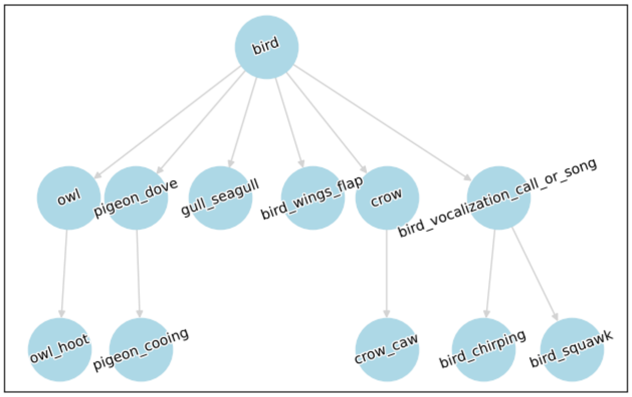


Figure 3: Example of standard label mapping for the standardized label *bird*.

An additional example, is illustrated in Listing 1, which involves retrieving the dataset labels mapped to the standard label *reverse_beeper*. The function returns a Python dictionary where dataset names serve as keys and their corresponding labels as values. This example highlights the package's feature to provide detailed and well-organized information about class labels, which is essential for analysis and integration of data.

Furthermore, the package comes with extensive documentation including a tutorial notebook and practical examples to demonstrate all functionalities discussed in this section. The interested reader is referred to the corresponding repository for more information about `py-salt`.

3.2. Applications

SALT can serve diverse applications and use cases through its functionalities. The provided python library, facilitates exploration of mapped datasets, both individually and in combination. An interesting use case comprises the compilation of data from multiple datasets to compose new datasets or collections. This is achieved by the use of a series of methods provided in `py-salt`, that allows gathering the desired labels from specific datasets or domains of interest. For example, to develop an audio classifier specialized in the detection of emergency signals, all relevant labels from different datasets *e.g., AudioSet, SINGA:PURA, ESC-50, etc.* can be easily accessed through the standard label *alarm_signal*. Similarly, to develop an urban sound monitoring system, various dataset labels can be aggregated through a set of standard labels such as *vehicle*, *engine*, and *outdoor_urban_or_manmade* from datasets *MAVD-traffic, AudioSet, FSD50K* and *SONYC-UST*, respectively. To give another example, for a system targeting the detection of domestic sound events, labels such as *kitchen* (*i.e. kitchen sounds*), *bell* and *television* can be aggregated to create a specialized classifier for recognizing common household sounds. Figure 5 illustrates the significant benefits of label aggregation in augmenting the amount of data

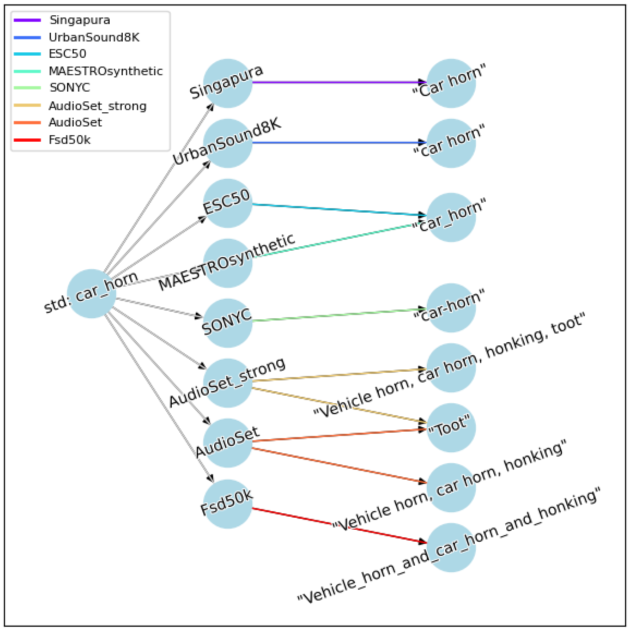


Figure 4: Example of dataset label mapping for the standardized label *car_horn*.

available for minority classes.

Another use case involves defining a common set of standard labels for cross-dataset evaluation purposes, *e.g.*, training on *SONYC* and testing on the same set of labels on *UrbanSound8K*. This approach is particularly useful for bench-marking audio analysis systems and assess their generalization capabilities. Overall, SALT can diminish inconsistencies and discrepancies between different datasets and promotes fair comparison of model performance.

```
from py_salt.event_mapping import EventExplorer()

# Init taxonomy explorer
e = EventExplorer()

# Get dataset mapping dictionary
e.get_mapping_for_std_label(’reverse_beeper’)

{’SONYC’: [’reverse-beeper’],
 ’Singapura’: [’Reverse beeper’],
 ’AudioSet_strong’: [’Reversing beeps’],
 ’AudioSet’: [’Reversing beeps’]}
```

Listing 1: Label search using the standardized label *reverse_beeper*

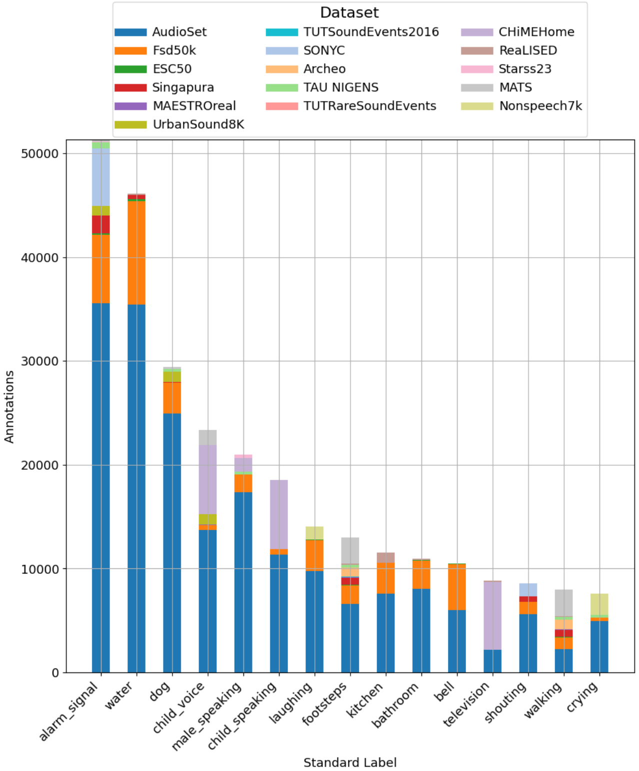


Figure 5: The benefit of label aggregation in selected standardized labels targeting domestic sound events.

4. CONCLUSION

In this paper, we introduced *SALT: Standardized Audio event Label Taxonomy* to unify existing sound taxonomies into a global one through the standardization of labels, while also addressing some of their limitations. Built upon AudioSet's hierarchical structure, SALT standardizes and extends labels across 24 environmental sound datasets, enhancing clarity and precision and enabling cross-dataset label compatibility. Furthermore, we support the use of SALT, by introducing a Python package that provides robust tools to perform cross-dataset label aggregation, explore hierarchical relationships and visualize label mappings. These capabilities streamlining data aggregation and analysis, make SALT a valuable resource for developing machine listening systems at scale.

5. ACKNOWLEDGMENT

This work was supported by the Audible project, funded by French BPI.

6. REFERENCES

[1] E. Vidaña-Vila, J. Navarro, D. Stowell, and R. M. Alsina-Pagès, “Multilabel acoustic event classification using real-world urban data and physical redundancy of sensors,” *Sensors*, vol. 21, no. 22, p. 7470, 2021.

[2] F. Angulo, S. Essid, G. Peeters, and C. Mietlicki, “Cosmopolite sound monitoring (cosmo): A study of urban sound event detection systems generalizing to multiple cities,” in *Proc. ICASSP*. IEEE, 2023, pp. 1–5.

[3] K. Suefusa, T. Nishida, H. Purohit, R. Tanabe, T. Endo, and Y. Kawaguchi, “Anomalous sound detection based on interpolation deep neural network,” in *Proc. ICASSP*. IEEE, 2020, pp. 271–275.

[4] G. Wichern, A. Chakrabarty, Z.-Q. Wang, and J. Le Roux, “Anomalous sound detection using attentive neural processes,” in *Proc. WASPAA*. IEEE, 2021, pp. 186–190.

[5] Y. Liu, J. Guan, Q. Zhu, and W. Wang, "Anomalous sound detection using spectral-temporal information fusion," in *Proc. ICASSP*. IEEE, 2022, pp. 816–820.

[6] J.-L. Durrieu, G. Richard, and B. David, "Singer melody extraction in polyphonic signals using source separation methods," in *Proc. ICASSP*. IEEE, 2008, pp. 169–172.

[7] Y. Ni, M. McVicar, R. Santos-Rodriguez, and T. De Bie, "An end-to-end machine learning system for harmonic analysis of music," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 6, pp. 1771–1783, 2012.

[8] S. Oramas, F. Barbieri, O. Nieto Caballero, and X. Serra, "Multimodal deep learning for music genre classification," *Transactions of the International Society for Music Information Retrieval. 2018; 1 (1): 4-21.*, 2018.

[9] S. Toshniwal, T. N. Sainath, R. J. Weiss, B. Li, P. Moreno, E. Weinstein, and K. Rao, "Multilingual speech recognition with a single end-to-end model," in *Proc. ICASSP*. IEEE, 2018, pp. 4904–4908.

[10] V. Pratap, A. Hannun, Q. Xu, J. Cai, J. Kahn, G. Synnaeve, V. Liptchinsky, and R. Collobert, "Wav2letter++: A fast open-source speech recognition system," in *Proc. ICASSP*. IEEE, 2019, pp. 6460–6464.

[11] D. Palaz, M. M. Doss, and R. Collobert, "Convolutional neural networks-based continuous speech recognition using raw speech signal," in *Proc. ICASSP*. IEEE, 2015, pp. 4295–4299.

[12] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *Proc. ACM Multimedia*, 2014, pp. 1041–1044.

[13] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. ICASSP*. IEEE, 2017, pp. 776–780.

[14] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "Fsd50k: an open dataset of human-labeled sound events," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 30, pp. 829–852, 2021.

[15] M. Cartwright, A. E. M. Mendez, A. Cramer, V. Lostanlen, G. Dove, H.-H. Wu, J. Salamon, O. Nov, and J. Bello, "Sonyc urban sound tagging (sonyc-ust): A multilabel dataset from an urban acoustic sensor network," 2019.

[16] P. Zinemanas, P. Cancela, and M. Rocamora, "Mavd: a dataset for sound event detection in urban environments," 2019.

[17] K. Ooi, K. N. Watcharasupat, S. Peksi, F. A. Karnapi, Z.-T. Ong, D. Chua, H.-W. Leow, L.-L. Kwok, X.-L. Ng, Z.-A. Loh, *et al.*, "A strongly-labelled polyphonic dataset of urban sounds with spatiotemporal context," in *Proc. APSIPA ASC*. IEEE, 2021, pp. 982–988.

[18] N. Turpault, R. Serizel, A. P. Shah, and J. Salamon, "Sound event detection in domestic environments with weakly labeled data and soundscape synthesis," in *Proc. DCASE*, 2019.

[19] C. Guastavino, "Everyday sound categorization," *Computational analysis of sound scenes and events*, pp. 183–213, 2018.

[20] W. W. Gaver, "What in the world do we hear?: An ecological approach to auditory event perception," *Ecological psychology*, vol. 5, no. 1, pp. 1–29, 1993.

[21] C. Guastavino, "Categorization of environmental sounds." *Canadian Journal of Experimental Psychology/Revue canadienne de psychologie expérimentale*, vol. 61, no. 1, p. 54, 2007.

[22] R. M. Schafer, *Our sonic environment and the soundscape: The tuning of the world*. Destiny Books, 1994.

[23] D. Moffat, D. Ronan, J. D. Reiss, *et al.*, "Unsupervised taxonomy of sound effects," *context*, vol. 6, no. 7, 2017.

[24] M. B. Cartwright, J. Cramer, A. E. M. Méndez, Y. Wang, H.-H. Wu, V. Lostanlen, M. Fuentes, G. Dove, C. Mydlarz, J. Salamon, O. Nov, and J. P. Bello, "Sonyc-ust-v2: An urban sound tagging dataset with spatiotemporal context," in *Proc. DCASE*, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:221641055

[25] K. J. Piczak, "Esc: Dataset for environmental sound classification," in *Proc. ACM Multimedia*, 2015, pp. 1015–1018.

[26] F. Angulo, S. Essid, G. Peeters, and C. Mietlicki, "Cosmopolite sound monitoring (cosmo): A study of urban sound event detection systems generalizing to multiple cities," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[27] R. M. Bittner, M. Fuentes, D. Rubinstein, A. Jansson, K. Choi, and T. Kell, "mirdata: Software for reproducible usage of datasets." in *ISMIR*, 2019, pp. 99–106.

[28] M. Fuentes, J. Salamon, P. Zinemanas, M. Rocamora, G. Paja, I. R. Román, M. Miron, X. Serra, and J. P. Bello, "Soundata: A python library for reproducible use of audio datasets," *arXiv preprint arXiv:2109.12690*, 2021.

MACHINE LISTENING IN A NEONATAL INTENSIVE CARE UNIT

Modan Tailleur[1]*, *Vincent Lostanlen*[1]†, *Jean-Philippe Rivière*[1], *Pierre Aumond*[2]

[1] Nantes Université, École Centrale Nantes, CNRS, LS2N, UMR 6004, F-44000 Nantes, France
[2] Université Gustave Eiffel, CEREMA, UMRAE, F-44344 Bouguenais, France

ABSTRACT

Oxygenators, alarm devices, and footsteps are some of the most common sound sources in a hospital. Detecting them has scientific value for environmental psychology but comes with challenges of its own: namely, privacy preservation and limited labeled data. In this paper, we address these two challenges via a combination of edge computing and cloud computing. For privacy preservation, we have designed an acoustic sensor which computes third-octave spectrograms on the fly instead of recording audio waveforms. For sample-efficient machine learning, we have repurposed a pretrained audio neural network (PANN) via spectral transcoding and label space adaptation. A small-scale study in a neonatological intensive care unit (NICU) confirms that the time series of detected events align with another modality of measurement: i.e., electronic badges for parents and healthcare professionals. Hence, this paper demonstrates the feasibility of polyphonic machine listening in a hospital ward while guaranteeing privacy by design.

Index Terms— Computational environmental audio analysis, edge computing, machine learning methods, privacy.

1. INTRODUCTION

Sound is a reliable and non-invasive carrier of information about human health [1]. Yet, historically, the subfield of medical acoustics has mainly focused on analyzing sounds as *produced* by patients: stutter [2], crackles [3], cough [4], and so on. Much less is known about the sounds as *heard* by patients in a clinical setting: as experimental psychologists have pointed out, the detailed description of acoustic events in intensive care units (ICU's) is typically overlooked in favor of sound pressure level measurements (SPL) [5]. Meanwhile, exposure to anthropogenic noise at unsafe SPL levels is known to induce stress, cognitive impairment and sleep disorders in children [6] and adults [7], thus calling for urgent remediation.

The case of neonatal intensive care units (NICU's), where premature babies receive special care to grow and survive, presents an even greater gap in research than adult ICU's [8]. During their time in the NICU, preterm infants are exposed to unpredictable sensory stimuli while undergoing a protracted period of rapid brain growth, causing lasting effects on cognitive ability [9]. Unfortunately, the auditory physiology and cognition of neonates have received insufficient attention from scientists until recently [10].

What is known with certainty is that parents have an essential role to play in the development of their newborn babies [11]. Indeed, an approach sometimes described as "kangaroo care" involves prolonged periods of skin-to-skin contact between the baby and either of the two parents, in addition to incubator placement. Promoting this approach requires to take the well-being of parents into consideration so that they feel included into collective care work.

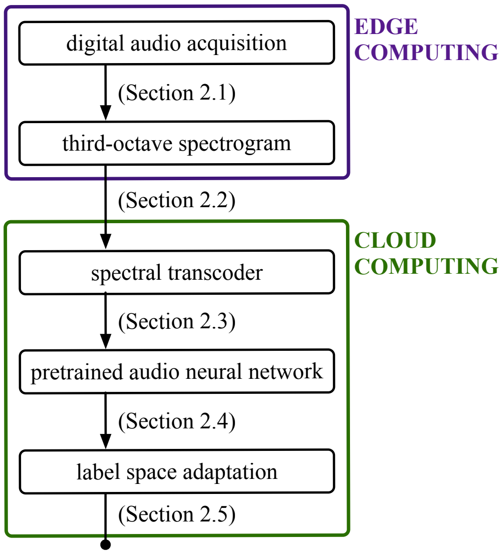


Figure 1: Flowchart of stages in the proposed approach. The first two stages are performed "on the edge". The last three stages are performed "on the cloud", i.e., on a central server.

For this purpose, we have launched a project on "listening to family experiences in the neonatological ward", or LIFEWARD for short. Here, the word "listening" is understood as both qualitative and quantitative: i.e., as enacted by interviews with parents as well as autonomous acoustic sensors. Although there is scientific consensus around the value of semi-structured interviews for neonatology—see, for example, [12]—the same cannot be said about machine listening. This is for at least three reasons. First, the deployment of acoustic sensors in a hospital raises pressing concerns about privacy preservation and cybersecurity. Secondly, the application of machine learning to the NICU is not straightforward, for lack of annotated training data. Thirdly, and perhaps most fundamentally, machine listening systems have not yet demonstrated their ability to reconstruct objective information about social bonds in the NICU. Addressing these three challenges is necessary before envisioning the integration of machine listening instruments within the toolkit of patient experience research.

In this article, we present a proof of feasibility of machine listening for neonatology. Prior work in this domain has focused on a single class of sound event—namely, the spontaneous cries of preterm newborns [13]. Meanwhile, our system is a multilabel

*Thanks to ANR AIBY4 (ANR-20-THIA-0011) for funding.
†Thanks to ANR MuReNN (ANR-23-CE23-0007-01) for funding.

sound event detector for adult voices, footsteps, oxygenators, and alarm devices. Furthermore, the originality of our approach is that it integrates all aspects of machine listening, from digital audio acquisition to sound event detection, into a mixed pipeline that involves both edge computing and cloud computing.

Figure 1 decomposes our approach into five stages, described in Sections 2.1 through 2.5. To comply with standards of privacy and security, the LIFEWARD sensor does not store the acquired waveforms. Rather, it extracts a third-octave spectrogram on the fly; i.e., a coarse estimate of power spectral density over windows of duration $125\,\mathrm{ms}$. This, in turn, brings its own challenges for sound event detection, which typically requires finer spectrotemporal information. We address this challenge via a pretrained “spectral transcoder”; i.e., a deep neural network for nonuniform resampling the time–frequency domain. We pass the output of the spectral transcoder to a pretrained audio neural network (PANN), with AudioSet as its label space. Lastly, we use domain-specific knowledge to narrow down this taxonomy for the NICU.

Section 3 presents the result of an in-progress study at an NICU, and provides tentative evidence for the feasibility of the proposed approach. Indeed, neural network predictions appear to coincide with isolated sound events of interest (Section 3.1) as well as timestamps from a non-audio modality of human presence (Section 3.2).

2. METHODS

2.1. Acoustic sensor

Our acoustic sensor is a Raspberry Pi, inspired by previous work on urban noise monitoring [14]. It acquires audio from an external USB microphone, specifically, the micW i436. The i436 is an omnidirectional electret microphone with a capsule diameter of approximately 7 mm, in compliance with NF EN 61672 Class-2 standards. Its sensitivity and frequency response has been calibrated manually by the manufacturer. After digital–analog conversion, the sample rate is 32 kHz. Our sensor is powered by the grid and “airgapped”, i.e., physically isolated from the public Internet and from each other. This is to reduce the risk of malicious data access.

2.2. Third-octave spectrogram

We use fast Fourier transforms (FFT) to design a third-octave filterbank with bands ranging from 20 Hz to 12.5 kHz, in compliance with the ANSI S1.1-1986 and IEC 61260-1:2014 standards [15]. We extract the magnitude response of each filter over nonoverlapping subbands of duration 125 ms. These operations are implemented in the C language, compiled for the Raspberry Pi, and executed in real time. The result is stored incrementally on a nonvolatile memory (“SD card”).

A perceptual evaluation on twelve subjects has shown that the third-octave spectrogram does not contain sufficient information to recover intelligible speech, at least via classical signal processing techniques—namely, Moore-Penrose pseudoinverse and Griffin-Lim algorithm for phase retrieval [15]. Thus, the third-octave spectrogram representation can be said to be privacy-aware, in the sense it mitigates the severity of a security breach should the SD card were to be lost or stolen in the healthcare facility.

Another advantage of computing third-octave spectrograms on the edge resides in its bitrate: around 3.71 kilobytes per second (kbps). This is lower than MP3 (128–320 kbps) and lossless audio (around 1 Mbps). The bitrate of third-octave spectrograms translates to around 320 megabytes per day, or 117 gigabytes per year. Thus, a single SD card suffices to contain all the spectrogram data over a longitudinal survey spanning the full length of stay of the preterm infant at the NICU.

2.3. Spectral transcoder

Previous work in urban environments has shown the potential of the third-octave spectrogram as a feature for sound event classification, both in supervised and self-supervised scenarios [16]. Yet, this previous work is unapplicable in the context of the NICU, for lack of annotated training data. Furthermore, note that it would not be possible to launch our own annotation campaign because, as explained before, our sensors do not record audio. We propose to circumvent this problem by relying on a pretrained audio neural network (PANN) for multilabel sound event detection and classification [17].

Here, a second issue arises: PANN does not operate upon the third-octave spectrogram but on a mel-frequency spectrogram, which has a finer temporal resolution (hop size of 10 ms) and a finer spectral resolution (64 bins on the mel scale). In principle, the required change of resolution could be achieved by a linear nonuniform resampler. Yet, in practice, this produces a blurry time–frequency representation which is not recognized by PANN as containing any events of interest. Against this issue, a deep neural network was developed by Tailleur et al. [18], which we call *spectral transcoder*, so as to recover a plausible mel-frequency spectrogram from a third-octave spectrogram measurement.

The spectral transcoder is a convnet with six layers. It is trained on TAU Urban Acoustic Scenes 2020 Mobile dataset [19] in a “teacher–student” scenario. The teacher is the composition of mel-frequency spectrogram and PANN whereas the student is the composition of third-octave spectrogram, spectral transcoder, and PANN. In other words, the spectral transcoder is not trained to minimize its mean square error with the mel-frequency ground truth (as a linear model would) but to generate a mel-frequency spectrogram whose spectrotemporal content has the same distribution of sound events as the ground truth. The training process involves minimizing a binary cross-entropy loss, computed between the PANN output of the student and that of the teacher, by updating solely the transcoder’s parameters. This is a kind of super-resolution procedure in which the implicit knowledge about the spectrotemporal characteristics of natural audio sounds is distilled from PANN into the spectral transcoder under the form of convnet weights. We refer to [18] for more details on the spectral transcoder.

2.4. AudioSet classification with PANN

Our PANN of choice is a residual network with 38 layers, or ResNet38 for short. It contains around 74M parameters. To this day, it is regarded as one of the most accurate general-purpose multilabel audio classifier among those which take the mel-frequency spectrogram as input. The PANN is trained on AudioSet, a dataset which contains over 2M 10-second audio clips which were extracted from YouTube videos. In the next sections, we refer to the composition of pretrained spectral transcoder and PANN as “PANN-1/3oct” model. We refer to [17] for further details on PANN.

Since PANN is a multilabel classifier, its output vector is unnormalized. For the sake of visualization, we have found it beneficial to rank predictions in decreasing order, normalize rank by the number of classes, and apply an inverse power transform. Given predictions $\boldsymbol{x}[k]$ for each class k, this procedure yields the α-compressed reciprocal rank

$$\boldsymbol{y}[k] = \left(\frac{K}{\sigma^{-1}[k]}\right)^{\alpha}, \tag{1}$$

where k is the class index, K is the total number of classes, σ is the sorting permutation such that $(\boldsymbol{x} \circ \sigma)[1] > \ldots > (\boldsymbol{x} \circ \sigma)[K]$, and $\alpha < 1$ is a constant exponent. We set $\alpha = 0.5$ in this paper.

2.5. Label space adaptation

The PANN-1/3oct model analyzes a third-octave spectrogram snippet of duration equal to 10 seconds and returns a vector of dimension 527, corresponding to the classes in AudioSet dataset. These classes are a subset of the 623-class AudioSet ontology[1], which has been defined by a Google Research team after scraping web-scale text data for “Hearst patterns”, i.e., of either of these forms [20]:

> [...] sounds such as X or Y [...]
> [...] X, Y, and other sounds [...]

This approach has proven fruitful for general-purpose audio classification: we refer to [21] for a review. Yet, it is unsuitable for the ICU, whose distribution of sound events is inadequately represented by textual mentions of sound events on the web. At the same time, training a classifier from scratch on a new taxonomy is out of the question for reasons of privacy preservation, as explained earlier.

Instead, we simply run the PANN-1/3oct model on third-octave spectrogram data from the NICU and look for some frequently occuring AudioSet classes. We find four activity patterns of interest: “conversation”, “walk, footsteps”, “train”, and “electronic music”. Although the former two sound events are plausible, the latter two are clearly not. Yet, after interviewing NICU employees, we may hypothesize that they yield indirect information: i.e., that “train” actually corresponds to the rumble of the oxygenator while “electronic music” corresponds to the ringtone of the hospital phone. We summarize this correspondence in Table 1.

Neonatal Intensive Care Unit (NICU)	AudioSet
Conversation	Conversation
Footsteps	Walk, footsteps
Oxygenator	Train
Hospital phone	Electronic music

Table 1: Mapping of sound event labels from the neonatal intensive care unit (NICU) to AudioSet.

3. APPLICATION

3.1. Deployment in a neonatal intensive care unit

Since 2018, a design company[2] have been partnering with Nantes University hospital and a nonprofit organization[3] to enhance the inclusion of parents in the NICU. The nonprofit organization collaborated with designers to refurnish a care room so as to facilitate the presence of parents alongside their newborn. In this context, the LIFEWARD sensor has offered the necessary guarantees for a safe and privacy-aware deployment in the NICU. We have obtained the approval of an ethical review board to deploy this sensor[4]. Six families have given their informed consent to participate in the LIFEWARD study: three in the aforementioned redesigned room and three in a standard room. The length of stay is approximately 90 days for each family. Thus, we have collected third-octave spectrogram data over 18 cumulated months.

3.2. Visualization of sound events

We now collect a few waveform-domain samples from the sound events of interest in a real NICU environment. This data collection stage is carried out with a handheld device, over short durations, and with the collaboration of NICU professionals. Specifically, we ring various kinds of alarms, activate oxygenators and other pumps, stomp our feet, and so forth. Admittedly, these sounds are too few to offer an independent quantitative evaluation of PANN-1/3oct: we refer to [18] for that matter. Still, they may serve as suggestive evidence for the fact that the correspondences which we hypothesized in Table 1 are adequate and useful in practice.

Figure 2 illustrates our findings for each of the four classes of interest. For example, we notice vertical patterns of high energy in the recording of footsteps, versus horizontal patterns in the recordings of oxygenator. These simple observations corroborate the prediction of the PANN-1/3oct model with label space adaptation: see Table 1. Those examples demonstrate one of the advantages of using the transcoder: one can double-check model predictions by displaying the transcoded spectrogram, despite not being able to listen to the underlying audio waveform.

3.3. Proof of feasibility for continuous monitoring

The previous section has confirmed the interest of the PANN-1/3oct model in the context of isolated sounds from the NICU, as acquired by a handheld device. It remains to be seen if this model remains informative in a real-world polyphonic context, as acquired by the LIFEWARD sensor. For this purpose, we propose to compare the detected events with another modality of measurement: i.e., electronic badges worn by parents and healthcare professionals. Via near-field communication (NFC), these badges yield information about who is present in the care room at any given time. Hence, they offer indirect confirmation for the feasibility of machine listening in the NICU, while remaining non-invasive and privacy-aware.

Figure 3 shows an example of PANN-1/3oct predictions from our real-world NICU dataset, together with timestamps from electronic badges. We notice that segments during which two adults are present in the room coincide with a rise in the presence of conversation—and, to a lesser degree, of footsteps. Meanwhile, the lowest values for the “conversation” class correspond to segments in which only one adult is present in the room. Yet, we recognize that these are only anecdotal observations. Future research is needed to expand the comparison of acoustical and non-acoustical information to a larger scale; i.e., multiple days and multiple rooms.

4. CONCLUSION

The DCASE community has a key role to play at the intersection between sound design and healthcare. Yet, fulfilling this role comes with challenge of its own, such as: privacy, cybersecurity, and limited labeled data. In this article, we have presented a first prototype

[1] Link to complete list of classes in the AudioSet dataset: https://research.google.com/audioset/dataset/index.html

[2] Sensipode

[3] the B.E.R.S.E association

[4] Groupe Nantais d’Éthique dans le Domaine de la Santé (GNEDS) - n°22-09-090

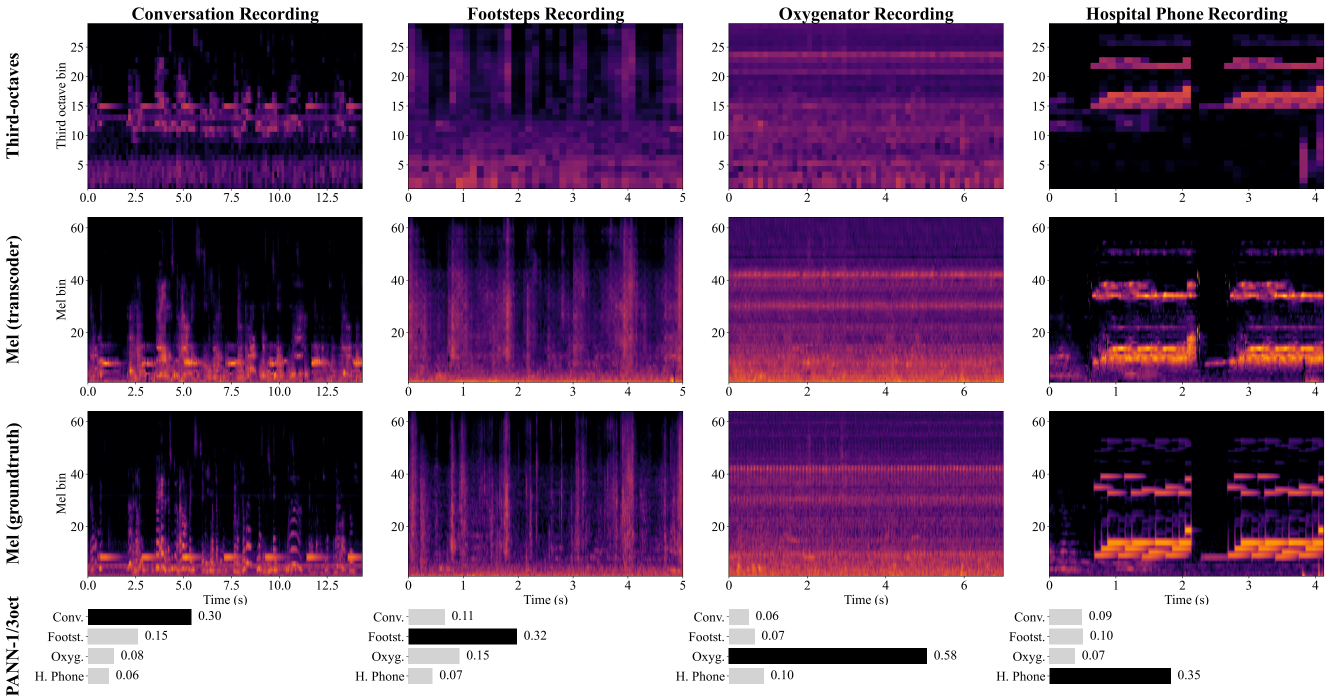


Figure 2: Spectrograms from audio recordings in the Neonatal Intensive Care Unit (NICU). First row corresponds to audio recordings transformed into fast third-octaves spectrograms. Second row corresponds to Mel spectrograms transcoded with the transcoder. Third row corresponds to groundtruth Mel spectrograms, obtained with Mel transformation on the waveform. PANN-1/3oct predictions, using the mapping between Audioset classes and NICU classes, are shown in fourth row.

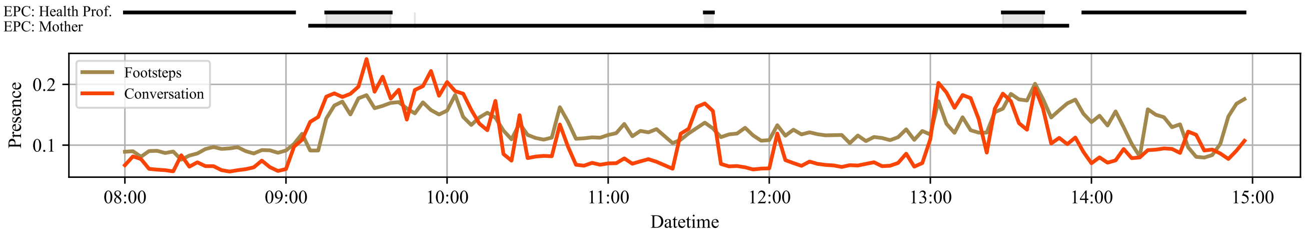


Figure 3: Presence of conversation and footsteps on a day of April 2023 in one room, as averaged over three-minute intervals. The badge of the health professional (EPC) and of the mother are also shown during the period. The shaded areas denote intervals in which more than one adult is present in the room.

of acoustic sensor which demonstrates the feasibility of sound event detection in a neonatal intensive care unit (NICU). The main limitation of our study is that, because our sensor does not record audio as waveforms, it is impossible to establish a “ground truth” by expert annotation. We have circumvented this limitation in two way: first, by evaluating the system on well-controlled isolated sounds; and second, by matching the sequence of detected sound events with non-acoustical information. In the future, we plan to refine the integration of multiple data modalities towards a more comprehensive understanding of patients and their lived experiences.

5. ACKNOWLEDGMENT

M.T. and V.L. thank Khrystyna Povkh for administrative management. V.L. thanks Sandie Cabon, Mathieu Lagrange, and Florence Levé for their insights. JP.R. thanks Ouest Industries Créatives, for their financial support (grant: OIC-2021-PEEL); and to Sensipode, the B.E.R.S.E. association, the STid Group, Didier Poiraud, Nicolas Fortin, Dr. Jean-Baptiste Muller, Judikaëlle Jacquin, Yannick Prié, as well as all the patients and hospital staff members for their valuable insights.

6. REFERENCES

[1] K. W. Beach and B. Dunmire, "Medical acoustics," in *Springer Handbook of Acoustics*. Springer, 2014, pp. 877–937.

[2] S. A. Sheikh, M. Sahidullah, F. Hirsch, and S. Ouni, "Machine learning for stuttering identification: Review, challenges and future directions," *Neurocomputing*, vol. 514, pp. 385–402, 2022.

[3] R. X. A. Pramono, S. Bowyer, and E. Rodriguez-Villegas, "Automatic adventitious respiratory sound analysis: A systematic review," *PLOS ONE*, vol. 12, no. 5, p. e0177926, 2017.

[4] K. S. Alqudaihi, N. Aslam, I. U. Khan, A. M. Almuhaideb, S. J. Alsunaidi, N. M. A. R. Ibrahim, F. A. Alhaidari, F. S. Shaikh, Y. M. Alsenbel, D. M. Alalharith, H. M. Alharthi, W. M. Alghamdi, and M. S. Alshahrani, "Cough Sound Detection and Diagnosis Using Artificial Intelligence Techniques: Challenges and Opportunities," *IEEE Access*, vol. 9, pp. 102 327–102 344, 2021.

[5] J. Mackrill, "Experiencing the hospital ward soundscape: Towards a model," *Journal of Environmental Psychology*, vol. 36, pp. 1–8, 2013.

[6] A. L. Bronzaft, "Supporting healthier urban environments with a sound and noise curriculum for students," *Cities & health*, vol. 5, no. 1-2, pp. 118–121, 2021.

[7] E. Murphy and E. A. King, *Environmental noise pollution: Noise mapping, public health, and policy*. Elsevier, 2022.

[8] S. Lenzi, S. Spagnol, and E. Özcan, "Improving the quality of the acoustic environment in neonatal intensive care units: A review of scientific literature and technological solutions," *Frontiers in Computer Science*, vol. 5, 2023.

[9] L. Gray and M. K. Philbin, "Effects of the neonatal intensive care unit on auditory attention and distraction," *Clinics in perinatology*, vol. 31, no. 2, pp. 243–260, 2004.

[10] M. K. Philbin, "The sound environments and auditory perceptions of the fetus and preterm newborn," *Early vocal contact and preterm infant brain development: Bridging the gaps between research and practice*, pp. 91–111, 2017.

[11] J. Baley, C. on Fetus, Newborn, K. Watterberg, J. Cummings, E. Eichenwald, B. Poindexter, D. L. Stewart, S. W. Aucott, K. M. Puopolo, and J. P. Goldsmith, "Skin-to-skin care for term and preterm infants in the neonatal icu," *Pediatrics*, vol. 136, no. 3, pp. 596–599, 2015.

[12] I. M. F. Medina, J. Granero-Molina, C. Fernández-Sola, J. M. Hernández-Padilla, M. C. Ávila, and M. d. M. L. Rodríguez, "Bonding in neonatal intensive care units: Experiences of extremely preterm infants' mothers," *Women and Birth*, vol. 31, no. 4, pp. 325–330, 2018.

[13] S. Cabon, B. Met-Montot, F. Porée, O. Rosec, A. Simon, and G. Carrault, "Automatic extraction of spontaneous cries of preterm newborns in neonatal intensive care units," in *2020 28th European Signal Processing Conference (EUSIPCO)*. IEEE, 2021, pp. 1200–1204.

[14] J. Ardouin, L. Charpentier, M. Lagrange, F. Gontier, N. Fortin, D. Ecotière, J. Picaut, and C. Mietlicky, "An innovative low-cost sensor for urban sound monitoring," in *Proceedings of Inter-Noise*, vol. 258, no. 5. Institute of Noise Control Engineering, 2018, pp. 2226–2237.

[15] F. Gontier, M. Lagrange, P. Aumond, A. Can, and C. Lavandier, "An efficient audio coding scheme for quantitative and qualitative large scale acoustic monitoring using the sensor grid approach," *Sensors*, vol. 17, no. 12, p. 2758, 2017.

[16] F. Gontier, V. Lostanlen, M. Lagrange, N. Fortin, C. Lavandier, and J.-F. Petiot, "Polyphonic training set synthesis improves self-supervised urban sound classification," *The Journal of the Acoustical Society of America*, vol. 149, no. 6, pp. 4309–4326, 2021.

[17] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "PANNs: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2880–2894, 2020.

[18] M. Tailleur, M. Lagrange, P. Aumond, and V. Tourre, "Spectral trancoder: Using pretrained urban sound classifiers on undersampled spectral representations," in *Proceedings of the International Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2023.

[19] A. Mesaros, T. Heittola, and T. Virtanen, "A multi-device dataset for urban acoustic scene classification," in *Proceedings of the International Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2018.

[20] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio Set: An ontology and human-labeled dataset for audio events," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal processing (ICASSP)*, 2017.

[21] T. Pellegrini, I. Khalfaoui-Hassani, E. Labbé, and T. Masquelier, "Adapting a ConvNeXt model to audio classification on AudioSet," in *Proceedings of the International Speech Association Conference (INTERSPEECH)*, 2023.

ACOUSTIC-BASED TRAFFIC MONITORING WITH NEURAL NETWORK TRAINED BY MATCHING LOSS FOR RANKING FUNCTION

Tomohiro Takahashi[1], *Natsuki Ueno*[1,2], *Yuma Kinoshita*[3], *Yukoh Wakabayashi*[4], *Nobutaka Ono*[1], *Makiho Sukekawa*[5], *Seishi Fukuma*[5], *Hiroshi Nakagawa*[5]

[1] Tokyo Metropolitan University, Tokyo, Japan
[2] Kumamoto University, Kumamoto, Japan
[3] Tokai University, Tokyo, Japan
[4] Toyohashi University of Technology, Aichi, Japan
[5] NEXCO-EAST ENGINEERING Company Limited, Tokyo, Japan

ABSTRACT

In this study, we propose an effective loss function for training neural networks (NNs) in acoustic-based traffic monitoring. This task involves estimating the number of vehicles from a fixed duration of acoustic input, such as one minute. Since the distribution of the number of passing vehicles depends on the road and can deviate significantly from a normal distribution, using Mean Square Error (MSE) as the loss function may not always lead to efficient learning. To address this, we introduce a matching loss for the ranking function into the loss function. This enhances learning by increasing the rank correlation between true and estimated vehicle counts. We evaluated the effectiveness of this loss function on the development dataset of the DCASE 2024 Challenge Task 10 under various input feature and network architecture conditions. The results demonstrate that the proposed loss function significantly improves Kendall's Tau Rank Correlation (KTRC) and Root Mean Square Error (RMSE), highlighting its potential for improving acoustic-based traffic monitoring systems.

Index Terms— matching loss, traffic monitoring, vehicle counting, deep neural network, acoustic sensing, microphone array

1. INTRODUCTION

Measuring road traffic conditions, including traffic volume, speed, density, time occupancy, vehicle type, and direction of travel, is essential for understanding real-time traffic situations. Traffic monitoring systems that provide this information to road traffic control systems and users are also crucial for smart city development [1,2]. Various sensors can be used for traffic monitoring, including intrusive systems embedded in the road (e.g., loop coils [3], vibration sensors), non-intrusive systems mounted over or on the side of the road (e.g., radar, cameras, infrared sensors, acoustic sensors), and off-road mobile devices (e.g., aircraft, satellites) [4].

Although acoustic sensors, that is, microphones installed on the roadside are not the most common method, they offer several advantages such as low installation and maintenance costs, low energy consumption, non-intrusiveness, and insensitivity to obstructions, shadows, and lighting conditions. Approaches to acoustic-based traffic monitoring are roughly classified into two types: rule-based [5–9] and machine-learning-based [10–19]. In particular, one of the latters, [18], provides a baseline system used in DCASE 2024 Challenge, the major international competition in the field of acoustic scene and event understanding. It represents the growing interest in recent years in machine-learning-based acoustic traffic monitoring. Various data augmentation methods have been proposed [20–29], especially, [18] investigates the effectiveness of data augmentation by an open-source road acoustic simulator [30].

In this study, following the DCASE 2024 Task 10 setup, we address the problem of counting the number of vehicles per vehicle type (car or Commercial Vehicle, CV) and per direction of travel (left or right) from acoustic signals. Specifically, we focus on the loss function for training a neural network (NN). Mean Square Error (MSE) is a common loss function for this task. However, using MSE may not be optimal for counting vehicles because the distribution of passing vehicles varies depending on the road and can significantly deviate from a normal distribution. Therefore, we propose a new loss function that aims to increase the rank correlation between the true and estimated number of vehicles since the rank correlation is a nonparametric measure and does not depend on the data distribution. This loss function is derived by applying the concept of matching loss [31] to the ranking function.

We conducted an experimental evaluation for checking the effectiveness of our loss function using training, validation, and synthetic data from the DCASE 2024 Challenge Task 10 development dataset [17]. We also compared several combinations of input acoustic features and two network architectures. Evaluated using Kendall's Tau Rank Correlation (KTRC) and Root Mean Square Error (RMSE), our loss function showed improvement in both metrics. Additionally, we assessed the estimation performance with and without pre-training. The results confirmed that pre-training enhanced the estimation performance.

This paper is organized as follows. Section 2 describes the input features and network architectures considered in this study of acoustic-based traffic monitoring with NN. Section 3 introduces the proposed loss function. Section 4 presents the experimental evaluation and results. Section 5 provides the conclusion.

2. INPUT FEATURES AND NETWORK ARCHITECTURES

In acoustic-based traffic monitoring with NN, the feature extraction from the input acoustic signal and the structure of the network architecture are crucial for efficient learning. In this section, we describe several input features and network architectures, whose effectivess has been confirmed in the baseline system of the DCASE 2024 Challenge Task 10 [17,18] and our previous work [19,20]. We will compare the performance of these various combinations in the experimental evaluation section 4.

2.1. Input features

Suppose that we have multiple microphones and let $\mathbf{x}_i \in \mathbb{R}^N$ be the acoustic signal of traffic sound captured by the ith microphone.

We express the short-time Fourier transform as

$$\mathbf{X}_i = \mathrm{STFT}(\mathbf{x}_i). \tag{1}$$

where $\mathbf{X}_i \in \mathbb{C}^{F\times T}$ is the complex-valued spectrogram with F frequency bins and T time frames. The amplitude of $\mathbf{X}_i$ and the phase difference between $\mathbf{X}_i$ and $\mathbf{X}_j$ are known to be the basis for effective features in acoustic-based traffic monitoring as follows.

LogMelSpec: $\mathbf{X}_i^{\mathrm{LMS}} \in \mathbb{R}^{M\times T}$ is calculated by taking the logarithm of the mel-scale transform of the amplitude spectrogram of $\mathbf{X}_i$, and M specifies the number of mel frequency bands.

LogPowSpec: $\mathbf{X}_i^{\mathrm{LPS}} \in \mathbb{R}^{F\times T}$, where the (f,t) element $\mathrm{X}_i^{\mathrm{LPS}}(f,t)$ is calculated by

$$\mathrm{X}_i^{\mathrm{LPS}}(f,t) = 10\log_{10}(|\mathrm{X}_i(f,t)|^2). \tag{2}$$

Here, f and t represent indices in the frequency and time directions, respectively, and $X_i(f,t)$ denotes the (f,t) element of $\mathbf{X}_i$ (the same notation applies to other matrices).

GCC-PHAT: $\mathbf{X}_{i,j}^{\mathrm{GCC}} \in \mathbb{R}^{G\times T}$, where the (τ,t) element $\mathrm{X}_{i,j}^{\mathrm{GCC}}(\tau,t)$ is calculated using the following equation [32]:

$$\mathrm{X}_{i,j}^{\mathrm{GCC}}(\tau,t) = \mathcal{F}^{-1}_{f\to\tau}\frac{X_i(f,t)X_j^*(f,t)}{|X_i(f,t)||X_j(f,t)|}. \tag{3}$$

Here, $\mathcal{F}^{-1}_{f\to\tau}$ is the inverse Fourier transform from f to τ. The indices i, j refer to distinct channels, and G specifies the number of GCC-PHAT coefficients.

PhaseDiff: $\mathbf{X}_{i,j}^{\mathrm{PDC}} \in \mathbb{R}^{F\times T}$ and $\mathbf{X}_{i,j}^{\mathrm{PDS}} \in \mathbb{R}^{F\times T}$, where the (f,t) elements, $\mathrm{X}_{i,j}^{\mathrm{PDC}}(f,t)$ and $\mathrm{X}_{i,j}^{\mathrm{PDS}}(f,t)$, are calculated using the following equations [33]:

$$\Delta\phi_{i,j}(f,t) = \arg(\mathrm{X}_i(f,t)/\mathrm{X}_j(f,t)), \tag{4}$$
$$\mathrm{X}_{i,j}^{\mathrm{PDC}}(f,t) = \cos(\Delta\phi_{i,j}(f,t)), \tag{5}$$
$$\mathrm{X}_{i,j}^{\mathrm{PDS}}(f,t) = \sin(\Delta\phi_{i,j}(f,t)). \tag{6}$$

2.2. Network architecture

Convolutional Neural Network (CNN)-based architecture is known to be effective in acoustic-based traffic monitoring. In this study, we consider the following two types of the CNN architectures.

CRNN: The amplitude-related and phase-related input features stack in the direction of the newly added channel dimension and are passed separately through network branches with the same structure. The network branches consist of convolutional encoders and Time-Distributed Multi-Layer Perceptrons (TD-MLP), composed of multiple Conv2D layers and a fully connected (FC) layer, respectively. TD-MLP is independently applied to each time frame of their input. Features that pass through each branch are concatenated and processed by a further TD-MLP layer, followed by a Gated Recurrent Unit (GRU) and an FC layer to regress labels.

ConvMixer: Each channel's amplitude-related and phase-related input features are concatenated in the direction of the frequency dimension and passed through the embedding layer, the ConvMixer layer [34, 35], and the classifier layer. In the embedding layer, patch embedding is applied to input features by handling short time frames in the input as patches. ConvMixer layer has a repeating network structure consisting of a pointwise convolution, which mixes the features for each time frame using an FC layer, and a depthwise convolution, which mixes the features by convolving in the direction of the time frame. In the classifier layer, the output of the ConvMixer layer is finally aggregated in the time-frame direction by the average pooling layer and then transformed into a one-dimensional vector representing the label by the FC layer.

3. PROPOSED LOSS FUNCTION

Let $y_k^{(*)}$ and $\hat{y}_k^{(*)} \in \mathbb{R}$ be the true and estimated vehicle counts for data k and label $(*) \in \{\text{car-l2r}, \text{car-r2l}, \text{CV-l2r}, \text{CV-r2l}\}$[1], respectively, where $k = 1,\dots,K$ is the data index within a batch of size $K \in \mathbb{N}$. Let $\mathbf{y}^{(*)}$ and $\hat{\mathbf{y}}^{(*)} \in \mathbb{R}^K$ be the collections of true and estimated counts, respectively, for all data $k = 1,\dots,K$.

A loss function is used to evaluate the performance of a model by quantifying the closeness between the true value $y_k^{(*)}$ and its estimate $\hat{y}_k^{(*)}$. One of the most common loss functions is the MSE such as

MSE:

$$L_{\mathrm{MSE}}(\hat{\mathbf{y}}^{(*)};\mathbf{y}^{(*)}) = \frac{1}{K}\sum_{k=1}^{K}(y_k^{(*)} - \hat{y}_k^{(*)})^2. \tag{7}$$

While, the key idea of our proposed loss function is to induce the correspondence between true and estimated order relations of data, aiming to enhance learning efficiently. Here, $\varphi : \mathbb{R}^K \to \mathbb{Z}^K$ is referred to as the ranking function, defined as

$$\varphi(\mathbf{y}^{(*)}) = \sum_{k=1}^{K}\begin{bmatrix}\mathrm{sign}(\hat{y}_1^{(*)} - \hat{y}_k^{(*)})\\ \vdots \\ \mathrm{sign}(\hat{y}_K^{(*)} - \hat{y}_k^{(*)})\end{bmatrix}. \tag{8}$$

Intuitively, $[\varphi(\mathbf{y}^{(*)})]_k$ denotes the number of elements smaller than $\hat{y}_k^{(*)}$ minus the number of elements larger than $\hat{y}_k^{(*)}$. Thus, φ maps the ranking of the input vector $\mathbf{y}^{(*)}$ into the integers within $\{-K+1,\dots,K-1\}$. If we want to bring $\varphi(\hat{\mathbf{y}}^{(*)})$ and $\varphi(\mathbf{y}^{(*)})$ closer, a straightforward loss function could be $||\varphi(\hat{\mathbf{y}}^{(*)}) - \varphi(\mathbf{y}^{(*)})||_2^2$, where $||\cdot||_2$ denotes the L_2 norm of a vector. However, this function cannot be used as a loss function due to the discontinuity and zero gradient of the function φ. Instead, we propose the following loss function.

Matching:

$$\begin{aligned}&L_{\mathrm{Matching}}(\hat{\mathbf{y}}^{(*)};\mathbf{y}^{(*)})\\ &= \frac{1}{K^2}\left(\frac{1}{2}\sum_{k=1}^{K}\sum_{l=1}^{K}|\hat{y}_k^{(*)} - \hat{y}_l^{(*)}| - \sum_{k=1}^{K}[\varphi(\mathbf{y}^{(*)})]_k\hat{y}_k^{(*)}\right).\end{aligned} \tag{9}$$

[1] The number of passenger cars or commercial vehicles (CVs) moving from left to right or right to left per minute.

Note that L_{Matching} is convex with respect to its first variable $\hat{\mathbf{y}}^{(*)}$, and its subgradient of L_{Matching} is provided by

$$\nabla L_{\mathrm{Matching}}(\hat{\mathbf{y}}^{(*)}; \mathbf{y}^{(*)}) = \frac{1}{K^2}\left(\varphi(\hat{\mathbf{y}}^{(*)}) - \varphi(\mathbf{y}^{(*)})\right). \qquad (10)$$

It means that minimizing L_{Matching} induces a correspondence between $\varphi(\hat{\mathbf{y}}^{(*)})$ and $\varphi(\mathbf{y}^{(*)})$, i.e., the ranking of the true and estimated data. A mathematical relationship like that between the loss function L_{Matching} and the vector-valued function φ is generally referred to as the matching loss [31]. Motivated by this concept, we incorporate the matching loss for the ranking function as described above to enhance the correspondence between the ranking of the true and estimated data.

Note that the ranking is invariant to bias, meaning that $\varphi(\hat{\mathbf{y}}^{(*)}) = \varphi(\hat{\mathbf{y}}^{(*)} + c\mathbf{1})$ for any constant c, where $\mathbf{1}$ denotes the vector each of whose element is one. This indicates that using only L_{Matching} is insufficient for estimating the correct number of vehicles without bias. Therefore, finally, we propose to combine these loss functions such as

MSE+Matching:

$$L_{\mathrm{MSE+Matching}} = (1-\lambda)L_{\mathrm{MSE}} + \lambda L_{\mathrm{Matching}}. \qquad (11)$$

where λ is a weight parameter for the combination.

4. EXPERIMENTAL EVALUATIONS

In this section, we present the conditions and results of an experiment to evaluate the effectiveness of our proposed loss function. The experiments were conducted using the input features and network architectures described in Section 2.

4.1. Experimental conditions

We trained our model using the training[2], validation[3], and synthetic[4] data from the DCASE 2024 Challenge Task 10 development dataset [17] and evaluated it using the validation data[5]. We referred to the baseline system of DCASE 2024 Challenge Task 10 [17, 18], using synthetic data exclusively for pre-training and real data for fine-tuning.

Two feature patterns were used as input features: **LogMelSpec+GCC-PHAT** and **LogPowSpec+PhaseDiff**, which combine amplitude-related and phase-related features. The number of channels is four $(i, j \in \{1, 2, 3, 4\})$, and the phase-related input features are computed between pairs of channels i and j. The sampling frequency F_s was 16 kHz, the STFT frame length N was 1024 points (64 ms), and the frameshift was 160 points (10 ms) for a 1-min signal. When the signal length $L = 60$ s, $F = \lfloor N/2 \rfloor + 1 = 513$ and $T = \lceil L \cdot F_s/(N/2) \rceil = 1875$. The number of frequency bands M of **LogMelSpec** was set to 48, and the number of coefficients G of **GCC-PHAT** was set to 96.

In the **CRNN**, we used six Conv2D layers of convolutional encoders, each with filters 32-32-64-64-128-128 and a kernel size of (5, 5) and a stride of 2 in both dimensions. The first TD-MLP has two layers with 128 neurons each, the second TD-MLP has three layers with 128 neurons each, the GRU has two layers with 128 neurons each, and the last FC layer has four neurons. In the **ConvMixer**, the number of embeddings was set to $T = 1875$, the feature vector dimension was 5, the kernel size of the depthwise convolution was 5, and the number of iterations was 5.

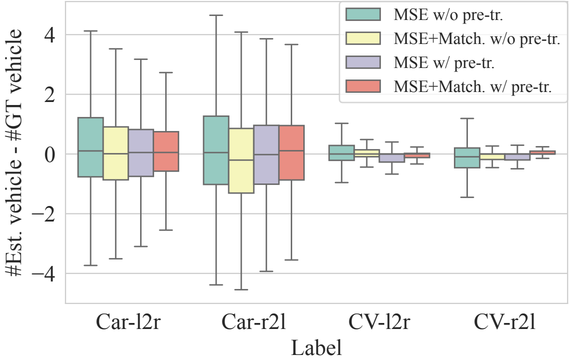


Figure 1: Error with and without **Matching** and pre-training

Two loss patterns were used as loss functions: **MSE** and **MSE+Matching**. During training, the average loss of each label was used to update the loss function, with the weighting coefficient λ of **MSE+Matching** set to 0.5, which was determined experimentally, and the batch size K set to 16. Additionally, the learning rate for pre-training and without pre-training learning was set to 0.00005, and the learning rate for fine-tuning was set to 0.0005, optimized by Adam [36]. We trained the model for 100 epochs and selected the best checkpoint based on validation loss. KTRC and RMSE were used as evaluation metrics and were evaluated for four labels: car-l2r, car-r2l, CV-l2r, and CV-r2l.

4.2. Experimental results

Table 1 shows the performance of vehicle counts for each location. **CRNN** performed better than **ConvMixer**, particularly in terms of CV accuracy and pre-training effectiveness. The estimation performance with pre-training, using **LogPowSpec+PhaseDiff**, **CRNN**, and **MSE+Matching**, was promising, particularly at location 6. Figure 1 shows the difference between true and estimated labels under these conditions, with and without **Matching** and pre-training. The results in the table and figure show that our proposed loss function and pre-training were confirmed to enhance estimation performance. Additionally, our proposed loss function improved not only in KTRC but also in RMSE. We submitted this best-performing condition system for the DCASE 2024 Challenge Task 10 [37].

5. CONCLUSIONS

In this study, we proposed a new loss function for acoustic-based traffic monitoring using NN. Our proposed loss function, derived from matching loss to the ranking function, aims to increase the rank correlation between the orders of true and estimated number of vehicles in each batch. We evaluated the effectiveness of our proposed loss function on the development dataset of the DCASE 2024 Challenge Task 10 and investigated good combinations of input acoustic features and network architectures. Our proposed loss function demonstrated improvements not only in KTRC but also in RMSE. As future work, our proposed loss function can be applied to other acoustics-based traffic monitoring tasks, such as traffic speed estimation.

[2] 7294 1-min training samples of real data collected from 6 sites.

[3] 7705 1-min validation samples of real data collected from 6 sites.

[4] 1224 1-min synthetic data generated via pyroadacoustics simulator [30].

[5] Since the labels for the DCASE 2024 Challenge Task 10 evaluation dataset are not yet publicly available, validation data were used [17].

Table 1: Performance of vehicle counts for each location

Loc.	Arc.	Input	Loss	Pre-tr.	↑ Kendall's Tau Rank Corr				↓ RMSE			
					car-l2r	car-r2l	CV-l2r	CV-r2l	car-l2r	car-r2l	CV-l2r	CV-r2l
1	CRNN	LogMelSpec+GCC-PHAT	MSE	✓	0.415	0.423	0.164	0.153	**2.619**	2.966	0.999	0.901
	CRNN	LogMelSpec+GCC-PHAT	MSE+Matching	✓	0.392	0.447	0.136	**0.172**	2.662	2.914	0.922	0.868
	CRNN	LogPowSpec+PhaseDiff	MSE	✓	0.39	**0.455**	**0.182**	0.129	2.689	**2.894**	0.88	0.884
	CRNN	LogPowSpec+PhaseDiff	MSE+Matching	✓	0.403	0.433	0.13	0.118	2.642	2.946	0.949	0.875
	ConvMixer	LogMelSpec+GCC-PHAT	MSE	✓	0.421	0.406	0.16	0.141	2.643	3.039	**0.837**	**0.835**
	ConvMixer	LogMelSpec+GCC-PHAT	MSE+Matching	✓	**0.429**	0.429	0.026	0.152	2.626	2.982	0.866	0.837
	ConvMixer	LogPowSpec+PhaseDiff	MSE	✓	0.29	0.306	0.121	0.105	2.894	3.23	0.842	0.84
	ConvMixer	LogPowSpec+PhaseDiff	MSE+Matching	✓	0.158	0.144	0.096	0.054	3.526	3.927	1.266	1.701
2	CRNN	LogMelSpec+GCC-PHAT	MSE	✓	0.768	0.409	0.201	0.026	**1.868**	2.627	0.815	0.678
	CRNN	LogMelSpec+GCC-PHAT	MSE+Matching	✓	0.685	0.376	0.086	-0.002	2.466	2.832	0.862	0.715
	CRNN	LogPowSpec+PhaseDiff	MSE	✓	0.685	0.462	-0.003	0.015	2.501	2.478	0.863	0.729
	CRNN	LogPowSpec+PhaseDiff	MSE+Matching	✓	**0.774**	**0.623**	0.128	**0.179**	1.9	**1.951**	0.824	**0.623**
	ConvMixer	LogMelSpec+GCC-PHAT	MSE	✓	0.415	0.361	**0.213**	-0.103	8.093	6.595	0.963	1.206
	ConvMixer	LogMelSpec+GCC-PHAT	MSE+Matching	✓	0.513	0.355	-0.118	0.045	4.384	3.429	1.766	1.003
	ConvMixer	LogPowSpec+PhaseDiff	MSE	✓	0.44	0.318	0.174	-0.126	3.536	3.194	**0.73**	0.919
	ConvMixer	LogPowSpec+PhaseDiff	MSE+Matching	✓	0.445	0.194	-0.193	-0.132	3.852	6.916	0.848	2.159
3	CRNN	LogMelSpec+GCC-PHAT	MSE	✓	0.545	0.578	**0.197**	**0.381**	1.739	1.281	0.3	**0.199**
	CRNN	LogMelSpec+GCC-PHAT	MSE+Matching	✓	0.548	**0.584**	0.081	-0.008	1.73	**1.275**	0.308	0.22
	CRNN	LogPowSpec+PhaseDiff	MSE	✓	**0.557**	**0.584**	0.191	0.226	**1.726**	1.286	**0.293**	0.224
	CRNN	LogPowSpec+PhaseDiff	MSE+Matching	✓	0.548	0.582	-0.028	-0.03	1.743	1.284	0.359	0.241
	ConvMixer	LogMelSpec+GCC-PHAT	MSE	✓	0.478	0.442	0.097	0.117	1.872	1.478	0.305	0.234
	ConvMixer	LogMelSpec+GCC-PHAT	MSE+Matching	✓	0.49	0.449	0.064	0.052	1.851	1.477	0.319	0.263
	ConvMixer	LogPowSpec+PhaseDiff	MSE	✓	0.494	0.486	0.079	0.141	1.845	1.43	0.3	0.218
	ConvMixer	LogPowSpec+PhaseDiff	MSE+Matching	✓	0.472	0.419	-0.044	0.008	1.87	1.514	0.298	0.215
4	CRNN	LogMelSpec+GCC-PHAT	MSE	✓	0.439	-0.013	-0.061	0.592	1.641	1.666	0.797	0.67
	CRNN	LogMelSpec+GCC-PHAT	MSE+Matching	✓	0.585	**0.467**	0.114	0.562	1.622	**0.801**	**0.501**	**0.41**
	CRNN	LogPowSpec+PhaseDiff	MSE	✓	**0.658**	-0.189	0.251	-0.197	**1.502**	2.16	0.667	0.57
	CRNN	LogPowSpec+PhaseDiff	MSE+Matching	✓	0.049	-0.013	-0.203	0.07	2.406	1.951	0.739	0.626
	ConvMixer	LogMelSpec+GCC-PHAT	MSE	✓	0.512	0.038	-0.266	-0.055	1.892	1.905	0.751	0.593
	ConvMixer	LogMelSpec+GCC-PHAT	MSE+Matching	✓	0.366	-0.164	0.301	-0.602	1.783	2.511	1.321	0.604
	ConvMixer	LogPowSpec+PhaseDiff	MSE	✓	0.341	0.29	**0.408**	**0.602**	9.199	1.699	1.097	1.702
	ConvMixer	LogPowSpec+PhaseDiff	MSE+Matching	✓	0.073	-0.215	0.301	0.456	2.048	1.355	0.592	0.917
5	CRNN	LogMelSpec+GCC-PHAT	MSE	✓	0.428	**0.498**	0.068	0.156	**0.771**	**0.619**	0.402	**0.187**
	CRNN	LogMelSpec+GCC-PHAT	MSE+Matching	✓	0.303	0.091	-0.063	**0.59**	0.827	0.886	0.374	0.234
	CRNN	LogPowSpec+PhaseDiff	MSE	✓	0.032	0.163	**0.157**	0.328	0.972	0.842	**0.352**	0.245
	CRNN	LogPowSpec+PhaseDiff	MSE+Matching	✓	**0.498**	0.283	-0.101	0.095	0.785	0.781	0.368	0.275
	ConvMixer	LogMelSpec+GCC-PHAT	MSE	✓	0.129	0.004	0.096	-0.134	1.013	0.889	0.372	0.407
	ConvMixer	LogMelSpec+GCC-PHAT	MSE+Matching	✓	-0.16	-0.149	-0.001	-0.107	1.271	0.842	0.769	0.278
	ConvMixer	LogPowSpec+PhaseDiff	MSE	✓	0.092	0.025	-0.048	0.135	0.947	0.852	0.357	0.284
	ConvMixer	LogPowSpec+PhaseDiff	MSE+Matching	✓	0.042	0.014	0.072	-0.034	0.947	1.006	0.372	0.28
6	CRNN	LogMelSpec+GCC-PHAT	MSE	✓	0.849	0.737	0.788	0.729	1.337	1.663	0.443	0.466
	CRNN	LogMelSpec+GCC-PHAT	MSE+Matching	✓	0.845	0.726	0.808	0.744	1.394	1.697	0.452	0.458
	CRNN	LogPowSpec+PhaseDiff	MSE	✓	0.827	0.713	0.753	0.681	1.507	1.748	0.519	0.511
	CRNN	LogPowSpec+PhaseDiff	MSE+Matching	✓	**0.854**	**0.738**	**0.821**	**0.761**	**1.288**	**1.607**	**0.433**	**0.451**
	ConvMixer	LogMelSpec+GCC-PHAT	MSE	✓	0.428	0.459	0.313	0.335	3.712	2.854	0.976	0.824
	ConvMixer	LogMelSpec+GCC-PHAT	MSE+Matching	✓	0.495	0.43	0.281	0.297	3.409	2.925	1.001	0.791
	ConvMixer	LogPowSpec+PhaseDiff	MSE	✓	0.494	0.49	0.404	0.308	3.368	2.764	0.955	0.798
	ConvMixer	LogPowSpec+PhaseDiff	MSE+Matching	✓	0.671	0.517	0.285	-0.069	2.584	2.707	1.002	0.857
6	CRNN	LogMelSpec+GCC-PHAT	MSE	—	**0.824**	0.709	0.78	**0.714**	**1.526**	**1.777**	0.514	**0.491**
	CRNN	LogMelSpec+GCC-PHAT	MSE+Matching	—	0.816	**0.71**	**0.788**	0.706	1.596	1.814	0.516	0.533
	CRNN	LogPowSpec+PhaseDiff	MSE	—	0.773	0.668	0.651	0.502	1.829	2.012	0.649	0.656
	CRNN	LogPowSpec+PhaseDiff	MSE+Matching	—	0.803	0.693	0.786	0.71	1.633	1.864	**0.509**	0.504
	ConvMixer	LogMelSpec+GCC-PHAT	MSE	—	0.752	0.659	0.252	0.141	1.968	1.998	1.005	0.835
	ConvMixer	LogMelSpec+GCC-PHAT	MSE+Matching	—	0.763	0.651	0.297	0.19	2.313	2.077	1.009	0.834
	ConvMixer	LogPowSpec+PhaseDiff	MSE	—	0.774	0.65	0.343	0.118	1.945	2.38	0.966	0.847
	ConvMixer	LogPowSpec+PhaseDiff	MSE+Matching	—	0.744	0.601	0.249	0.148	2.403	2.474	1.036	0.852

6. REFERENCES

[1] R. Du, P. Santi, M. Xiao, A. V. Vasilakos, and C. Fischione, "The sensable city: A survey on the deployment and management for smart city monitoring," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1533–1560, 2019.

[2] M. Won, "Intelligent traffic monitoring systems for vehicle classification: A survey," *IEEE Access*, vol. 8, pp. 73 340–73 358, 2020.

[3] B. Coifman and S. Neelisetty, "Improved speed estimation from single-loop detectors with high truck flow," *Journal of Intelligent Transportation Systems*, vol. 18, no. 2, pp. 138–148, 2014.

[4] P. T. Martin, Y. Feng, and X. Wang, "Detector technology evaluation," Mountain-Plains Consortium, MPC Report 03-154, 2003.

[5] M. A. Sobreira-Seoane, A. Rodríguez Molares, and J. L. Alba Castro, "Automatic classification of traffic noise," *The Journal of the Acoustical Society of America*, vol. 123, no. 5, pp. 3823–3823, 2008.

[6] G. Szwoch and J. Kotus, "Acoustic detector of road vehicles based on sound intensity," *Sensors*, vol. 21, no. 23, 7781 (18 pages), 2021.

[7] J. Kotus and G. Szwoch, "Estimation of average speed of road vehicles by sound intensity analysis," *Sensors*, vol. 21, no. 16, 5337 (18 pages), 2021.

[8] T. Toyoda, N. Ono, S. Miyabe, T. Yamada, and S. Makino, "Traffic monitoring with ad-hoc microphone array," in *Proceedings of the 14th International Workshop on Acoustic Signal Enhancement (IWAENC)*, pp. 318–322, 2014.

[9] S. Ishida, K. Mimura, S. Liu, S. Tagashira, and A. Fukuda, "Design of simple vehicle counter using sidewalk microphones," in *Proceedings of the ITS European Congress*, 10 pages, 2016.

[10] A. Y. Nooralahiyan, M. Dougherty, D. McKeown, and H. R. Kirby, "A field trial of acoustic signature analysis for vehicle classification," *Transportation Research Part C: Emerging Technologies*, vol. 5, no. 3-4, pp. 165–177, 1997.

[11] J. George, L. Mary, and K. S. Riyas, "Vehicle detection and classification from acoustic signal using ANN and KNN," in *Proceedings of the International Conference on Control Communication and Computing (ICCC)*, pp. 436–439, 2013.

[12] V. Tyagi, S. Kalyanaraman, and R. Krishnapuram, "Vehicular traffic density state estimation based on cumulative road acoustics," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1156–1166, 2012.

[13] J. Abeßer, S. Gourishetti, A. Kátai, T. Clauß, P. Sharma, and J. Liebetrau, "IDMT-traffic: An open benchmark dataset for acoustic traffic monitoring research," in *Proceedings of the 29th European Signal Processing Conference (EUSIPCO)*, pp. 551–555, 2021.

[14] S. Djukanović, Y. Patel, J. Matas, and T. Virtanen, "Neural network-based acoustic vehicle counting," in *Proceedings of the 29th European Signal Processing Conference (EUSIPCO)*, pp. 561–565, 2021.

[15] N. Bulatović and S. Djukanović, "Mel-spectrogram features for acoustic vehicle detection and speed estimation," in *Proceedings of the 26th International Conference on Information Technology (IT)*, 4 pages, 2022.

[16] T. Shinohara, Y. Wakabayashi, R. Scheibler, N. Ono, N. Aizawa, and H. Nakagawa, "Sound-based speed estimation using a neural network," in *Proceedings of the Spring meeting of the Acoustical Society of Japan*, pp. 385–386, 2020, (in Japanese).

[17] S. Ghaffarzadegan, L. Bondi, W.-C. Lin, A. Kumar, H.-H. Wu, H.-G. Horst, and S. Das, "Sound of traffic: A dataset for acoustic traffic identification and counting," in *Technical Report of the DCASE2024 Challenge*, 2024.

[18] S. Damiano, L. Bondi, S. Ghaffarzadegan, A. Guntoro, and T. van Waterschoot, "Can synthetic data boost the training of deep acoustic vehicle counting networks?" in *Proceedings of the 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 631–635, 2024.

[19] T. Takahashi, Y. Kinoshita, Y. Wakabayashi, N. Ono, J. Honda, S. Fukuma, A. Kitamori, and H. Nakagawa, "Acoustic traffic monitoring based on deep neural network trained by stereo-recorded sound and sensor data," in *Proceedings of the 31st European Signal Processing Conference (EUSIPCO)*, pp. 935–939, 2023.

[20] T. Takahashi, Y. Kinoshita, N. Ueno, Y. Wakabayashi, N. Ono, J. Honda, S. Fukuma, A. Kitamori, and H. Nakagawa, "Augmentation of various speed data by controlling frame overlap for acoustic traffic monitoring," in *Proceedings of Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 2068–2072, 2023.

[21] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Proceedings of the Interspeech*, pp. 2613–2617, 2019.

[22] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 13 001–13 008, 2020.

[23] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *Proceedings of the International Conference on Learning Representations*, 13 pages, 2018.

[24] S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo, and J. Choe, "CutMix: Regularization strategy to train strong classifiers with localizable features," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6022–6031, 2019.

[25] G. Kim, D. K. Han, and H. Ko, "SpecMix: A mixed sample data augmentation method for training with time-frequency domain features," in *Proceedings of the Interspeech*, pp. 546–550, 2021.

[26] H. Wang, Y. Zou, and W. Wang, "SpecAugment++: A hidden space data augmentation method for acoustic scene classification," in *Proceedings of the Interspeech*, pp. 551–555, 2021.

[27] H. Nam, S.-H. Kim, and Y.-H. Park, "Filteraugment: An acoustic environmental data augmentation method," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4308–4312, 2022.

[28] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "RandAugment: Practical automated data augmentation with a reduced search space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 3008–3017, 2020.

[29] J. Han, M. Matuszewski, O. Sikorski, H. Sung, and H. Cho, "Randmasking augment: A simple and randomized data augmentation for acoustic scene classification," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5 pages, 2023.

[30] S. Damiano and T. van Waterschoot, "Pyroadacoustics: a road acoustics simulator based on variable length delay lines," in *Proceedings of the 25th International Conference on Digital Audio Effects (DAFx22)*, pp. 216–223, 2022.

[31] D. Helmbold, J. Kivinen, and M. Warmuth, "Relative loss bounds for single neurons," *IEEE Transactions on Neural Networks*, vol. 10, no. 6, pp. 1291–1304, 1999.

[32] C. Knapp and G. Carter, "The generalized correlation method for estimation of time delay," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 4, pp. 320–327, 1976.

[33] Z.-Q. Wang, J. Le Roux, and J. R. Hershey, "Multi-channel deep clustering: Discriminative spectral and spatial embeddings for speaker-independent speech separation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5 pages, 2018.

[34] A. Trockman and J. Z. Kolter, "Patches are all you need?" *arXiv preprint arXiv:2201.09792*, 2022.

[35] R. Baidya and H. Jeong, "YOLOv5 with ConvMixer prediction heads for precise object detection in drone imagery," *Sensors*, vol. 22, no. 21, 8424 (17 pages), 2022.

[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations*, 15 pages, 2015.

[37] T. Takahashi, N. Ueno, Y. Kinoshita, Y. Wakabayashi, N. Ono, M. Sukekawa, S. Fukuma, and H. Nakagawa, "Neural network training with matching loss for ranking function," in *Technical Report of the DCASE2024 Challenge*, 2024.

TOWARDS LEARNING A DIFFERENCE-AWARE GENERAL-PURPOSE AUDIO REPRESENTATION

Daiki Takeuchi, Masahiro Yasuda, Daisuke Niizumi, Noboru Harada

NTT Corporation, Japan

ABSTRACT

General-purpose audio representations with self-supervised learning have shown promising results on diverse tasks. Methods such as BYOL-A try to learn semantically robust representation by ignoring differences between two data computed using data augmentations that simulate semantically similar data from the same input. However, some audio-difference-related tasks require representations that are sensitive to slight semantic differences while maintaining robustness to similar data. This study investigates how to learn difference-aware audio representations. We propose subtraction-consistent representation learning in which mixed sounds are separable by subtracting representations in latent space. In the proposed method, an additional network extending BYOL-A learns the difference between a sound sample and its down-mix with another sound sample. Experiments confirmed that the proposed method improves the accuracy of difference-aware audio tasks while maintaining the general-purpose audio representation performance.

Index Terms— general-purpose audio representation, audio difference, self-supervised learning

1. INTRODUCTION

General-purpose audio representations with self-supervised learning have shown promising results on diverse tasks [1–4]. Some of the self-supervised learning methods try to semantically robust learn representations by ignoring differences between two data augmentations applied to the same input. Data augmentations, such as time shifting, pitch shifting, and mixing other audio samples or noise, are designed and selected to emulate divisions to be ignored to obtain semantically similar representations in the latent space. As a result, learned representation will be robust to the difference between semantically similar data.

However, some difference-aware audio tasks, such as audio retrieval with auxiliary information [5], require representations that are sensitive to slight semantic differences while maintaining robustness to similar data. Existing general-purpose representation learning methods do not sufficiently solve this kind of task.

To address the lack of difference awareness in conventional self-supervised learning, we propose subtraction-consistent representation learning in which mixed sounds are separable by subtracting representations in latent space. The overview of the proposed method is shown in Fig. 1. The proposed method is implemented as an extension of BYOL-A [3]. Subtraction-consistent representation learning is based on the hypothesis that the semantic information present in a mixture of two sounds at similar sound pressure levels is equivalent to the combined semantic information of the two sounds before mixing. Our training method subtracts the representation of one mixed audio sample from the representation of the mixture and maximizes the agreement between the remaining representation of the subtraction and the representation of the other mixed audio sample. Multitask learning of BYOL-A and subtraction-consistent representation learning losses are performed during training. BYOL-A learns semantically robust audio representation, while subtraction-consistent representation learning makes that representation aware of differences. As a result, our method should learn a difference-aware general-purpose audio representation.

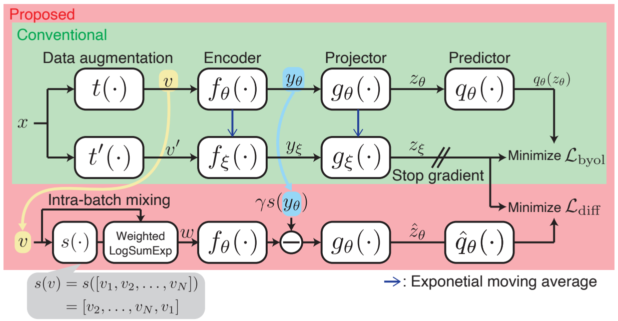


Figure 1: Overview of the proposed method and BYOL-A (conventional method). The proposed method (colored in red) extends BYOL-A (colored in green), mixing the augmented view v among other batch data to make a mixed input w. We train the proposed method to predict the BYOL-A target network output z_ξ from the difference between the encoder outputs of v and w.

Experiments confirm the learned representation by the proposed method improves the performance on two difference-aware audio tasks: environmental sound classification under noisy conditions and audio retrieval with auxiliary information. We also evaluate the learned representations in various downstream tasks and confirm that the performance was comparable to that learned by conventional BYOL-A. Therefore, the proposed method learns the difference-aware audio representation without degrading the general-purpose audio representation performance.

2. RELATED WORK

2.1. Self-supervised learning for audio representation

The general-purpose audio representation with self-supervised learning is effective for diverse tasks, including environmental sounds, music, and speech. BYOL-A [3] combines the self-supervised learning method Bootstrap Your Own Latent [6] (BYOL) with audio data augmentation. It learns representations invariant to differences in background noise and changes in the pitch and duration of audio. COLA [1] uses contrastive learning to learn representations that become closer to the segments cropped from the same audio clip and farther among the segments from the different audio clips, making the representations of an audio clip invariant to the cropping location. Fonseca et al. [2], and DeLoRes [4] also learn representations invariant to audio differences produced by data augmentation.

While they learn representations robust to changes produced by data augmentation and differences in segment cropping locations, they do not explicitly learn to encode information about differences in audio. This study investigates the learning of a general-purpose audio representation with awareness of audio differences by introducing the difference-based loss created by mixing sounds.

2.2. Difference-aware audio tasks

The recognition and retrieval tasks related to audio differences have also been studied. In [5], audio retrieval with auxiliary information was proposed. The content-based audio retrieval with text-query modifier [5] enables us to search an audio clip from an audio sample and the description of the difference. This method uses the common latent space between audio clips and descriptions of differences.

The methods to generate text explaining the difference between two sounds have also been studied [7, 8]. In [8], self-supervised learning focusing on the fact that input two audio clips are similar but slightly different is applied for learning the audio difference encoder. For the audio captioning system, the training method using the difference between the audio representation of before and after mixing is proposed in [9]. This study fixed the parameters of the encoder model that outputs acoustic representations and utilized the differences to train the text generation model. Unlike this study, we use differences to learn the parameters of the encoder model that outputs the audio representation.

3. BACKGROUND: BYOL-A

BYOL-A [3] is the method to obtain general-purpose audio representation by self-supervised training based on the BYOL framework [6]. The green area in Fig. 1 shows the overview of the BYOL training procedure. BYOL framework uses online and target networks with parameters θ and ξ, respectively. The online network has encoder f_θ, projector g_θ, and predictor q_θ. The target network has encoder f_ξ and projector g_ξ. The parameter of the target network ξ is the exponential moving average of the parameter of the online network θ. In the online network, compute v by data augmentation t to input x, then pass through the encoder, projector, and predictor to obtain $q_\theta(z_\theta)$. In the target network, compute v' by another data augmentation t' to input x, then pass through the encoder and projector to obtain z_ξ. After that, the normalized mean squared error of $q_\theta(z_\theta)$ and z_ξ is used for training loss:

$$\begin{aligned}\mathcal{L}_{\text{byol}} &= ||l_2(q_\theta(z_\theta)) - l_2(z_\xi)||_2^2 \\ &= 2 - 2 \cdot \frac{\langle q_\theta(z_\theta), z_\xi \rangle}{||q_\theta(z_\theta)||_2 \cdot ||z_\xi||_2},\end{aligned} \tag{1}$$

where $l_2(\cdot)$ is l_2-normalization, and $\langle x, y \rangle$ indicates the inner product of x and y. Thus, the BYOL framework can obtain the feature representation robust to the data augmentation t and t', and designing the data augmentation is one of the important elements to obtain better representation.

BYOL-A uses mel-spectrogram to preprocess the audio signal and three data augmentation methods that consider the nature of the audio signal: Mixup, random resize crop (RRC), and random linear fader (RLF). Mixup randomly adds another sound as background sound, RRC performs shifts and stretches in the axis of time and frequency randomly, and RLF makes random changes of temporal amplitude, which simulates fade in or out. BYOL-A applies Mixup, RRC, and RLF to input x sequentially and outputs the data-augmented views v and v'.

4. PROPOSED METHOD

The proposed method adds self-supervised learning to represent the relation between the audio signals before and after the mixture through differences in feature representations in the training procedure of BYOL-A. The training procedure of the proposed method is shown in red in Fig. 1. The proposed method is structured to include BYOL-A and in addition to a conventional loss $\mathcal{L}_{\text{byol}}$, it learns to predict the target network output z_ξ from the difference between audio representations before and after the mixture. The computational procedure of the proposed method branches from the input v after data augmentation, following the conventional BYOL-A. First, the mixture w is obtained by intra-batch mixing v with its index-shifting $s(v)$ and weighted log-sum-exp:

$$w = \log(\gamma \exp(s(v)) + (1-\gamma)\exp(v)), \tag{2}$$

where, γ is the mixing rate, s is the intra-batch shift operator, $s(v) = s([v_1, v_2, \ldots, v_N]) = [v_2, \ldots, v_N, v_1]$ and v_n is n-th data of v. Then, the difference between the encoder output of the mixture $f_\theta(w)$ and encoder output of the sound before mixing multiplied by the mixing ratio $\gamma s(y_\theta)$ is calculated and input into the projector g_θ and another predictor $\hat{q}_\theta$ to compute $\hat{q}_\theta(\hat{z}_\theta)$. Finally, we get the difference loss $\mathcal{L}_{\text{diff}}$, a normalized mean squared error between $\hat{q}_\theta(\hat{z}_\theta)$ and z_ξ:

$$\begin{aligned}\mathcal{L}_{\text{diff}} &= ||l_2(\hat{q}_\theta(\hat{z}_\theta)) - l_2(z_\xi)||_2^2 \\ &= 2 - 2 \cdot \frac{\langle \hat{q}_\theta(\hat{z}_\theta), z_\xi \rangle}{||\hat{q}_\theta(\hat{z}_\theta)||_2 \cdot ||z_\xi||_2}.\end{aligned} \tag{3}$$

The training step backpropagates the weighted sum of two loss $(1-\lambda)\mathcal{L}_{\text{byol}} + \lambda\mathcal{L}_{\text{diff}}$, where λ is the weight parameter.

5. EXPERIMENTS

We conducted the following experiments to evaluate the audio representations learned by the proposed method, and we used BYOL-A [3] as the baseline method.

5.1. Pre-training Setup

All audio data was transformed into a mel-spectrogram with a sampling frequency of 16,000 Hz, window size of 25 ms, hop size of 10 ms, and mel-spaced frequency bins $F = 64$ in the range of 50 to 8,000 Hz. The pre-training dataset was a random sample of 200,000 files from AudioSet [10]. Note that it was approximately 1/10 of the original size. The pre-training only utilized audio files without employing any labels. The same setup for data augmentation, exponential moving average, and model structures was used as the conventional method [3]. Adam [11] was used as the optimizer with a learning rate 0.001. The number of epochs was set to 100. The weight parameter λ, which decides the balance between $\mathcal{L}_{\text{byol}}$ and $\mathcal{L}_{\text{diff}}$ was set to 0, 0.1, 0.2, 0.5, or 0.8. Note that $\lambda = 0$ corresponds to the baseline method. The mixing rate of the intra-batch mixing γ is randomly sampled uniformly between 0.4 and 0.6 for each input.

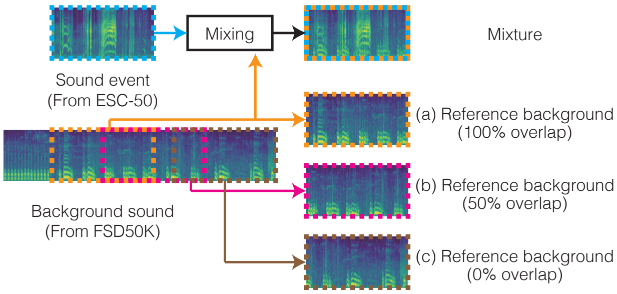


Figure 2: Procedure to generate BgKnown ESC-50. We mix the ESC-50 samples with the FSD50K sample as background noise to create a mixture and three reference background sounds.

Table 1: BgKnown ESC-50 results (%). A larger λ learned more from $\mathcal{L}_{\text{diff}}$ improves accuracy, validating that the proposed approach achieved the difference-aware property.

Method	λ	Mix	(a) 100%	(b) 50%	(c) 0%
Baseline	0	47.25	55.29	52.21	47.92
Proposed	0.1	47.39	55.54	52.46	48.50
Proposed	0.2	**47.42**	57.54	53.37	48.67
Proposed	0.5	47.33	58.96	54.63	50.50
Proposed	0.8	45.84	**59.96**	**55.50**	**51.96**

5.2. Evaluation: Background-known ESC-50

This experiment verified that the audio representation learned by the proposed method holds effective information about the audio differences for solving a task. To do so, we created a dataset, Background-known ESC-50 (BgKnown ESC-50), and tested the pre-trained models.

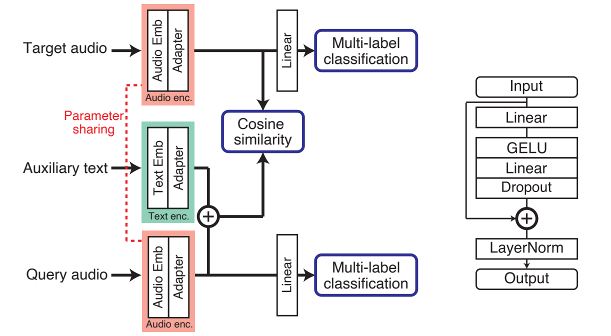
Dataset: Background-known ESC-50

BgKnown ESC-50 extends ESC-50 [12], an environmental sound classification task with 50 classes, by mixing the FSD50K audio files as background noise to the ESC-50 audio files. As shown in Fig. 2, we created a mixture (an ESC-50 audio contaminated with noise) and three reference backgrounds. While solving a task using only the mixture is challenging due to the noise, we made one of the reference backgrounds available; the more effectively the solver utilizes the difference between a mixture and a reference, the higher the task performance.

We randomly selected the FSD50K sample with 10 seconds or longer and cropped (a) a 5-second long clip, (b) a 5-second long clip with 50% overlap with (a), and (c) a 5-second long clip without overlap from (a), and mixed (a) into ESC-50 sample with a random SNR between 0 to 3 dB using Scaper [13]. We kept the labels unchanged. Among the split folds of ESC-50, we assigned 1, 2, and 3 to the training set (1200 files) and 4 and 5 to the test set (800 files). We used the FSD50K development and evaluation sets as the background noise for the training and test sets, respectively.

Experimental setup

We conducted a linear evaluation using feature differences on BgKnown ESC-50. First, we used the pre-trained encoder f_θ to obtain representations y_{mix} and y_{bg} of the mixture and reference background and obtained the difference representation $y_{\text{diff}} = y_{\text{mix}} - y_{\text{bg}}$. Then, we conducted a linear evaluation using the y_{diff} on the three problem settings (a) to (c).



(a) Model structure for audio retrieval with auxiliary information (b) Structure of adapter

Figure 3: Model and adapter structure for audio retrieval with auxiliary information. We train the system using a contrastive learning and classification task. Audio Emb and Text Emb indicate audio and text embedding layers, respectively. GELU is the Gaussian error linear unit [14].

Table 2: APwD-Dataset results (%). The proposed method improves the audio encoder, performing better than the conventional and baseline, with the best results using λ of 0.2 to 0.5.

		Rain			**Traffic**		
Method	λ	**R@1**	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**
Conventional [5]	-	44.5	72.1	76.9	39.1	62.2	69.5
Baseline	0	50.23	71.66	75.26	36.86	58.59	67.93
Proposed	0.1	51.96	71.63	74.83	37.96	59.73	66.9
Proposed	0.2	**53.99**	**72.06**	**76.00**	37.70	60.06	68.00
Proposed	0.5	52.76	71.73	75.73	**39.73**	60.63	**68.30**
Proposed	0.8	51.66	70.63	74.86	39.56	**61.36**	68.16

We followed the standard linear evaluation procedure in the conventional method [3] that trains a single linear layer, taking the difference representation y_{diff} as input. We set the training epochs for 200 with early stopping based on the validation loss value, assigned 10% of the training set as the validation set, and used the Adam optimizer with a learning rate of 0.001. We ran the experiments with different random seeds three times and averaged the results.

Results

Table 1 shows the results of BgKnown ESC-50. In addition to the (a) to (c), we also tested y_{mix} as is in the linear evaluation, denoted as "Mix". The results show that the proposed method improved accuracy with larger λ for the (a) to (c) when using the difference representation y_{diff}. In contrast, the results stayed around 47% for the Mix when we used the representation of the y_{mix} as it is instead of y_{diff}. These results demonstrate that the representation of the proposed method holds effective information about the audio differences.

Notably, the (c) 0% results show improvement despite no direct overlap with the mixed background noise. The segments cropped from the same audio clip share the background sounds (or sound scene of the clip), indicating that the information about the audio difference represents the clip-level (or semantic-level) information of the audio clip.

5.3. Evaluation: Audio retrieval with auxiliary information

We validated the effectiveness of the difference-aware representation for the difference-aware audio task. We evaluated the represen-

Table 3: Linear evaluation results on audio classification tasks (%) with 95% CI. The results in bold are the best scores in each task. Many underlined results within the 95% confidence interval of the baseline show that our models maintain baseline performance.

Method	λ	ESC-50	US8K	SPCV2	VC1	VF	CRM-D	GTZAN	NSynth	Surge	Average
Baseline	0	82.70 ± 1.76	79.43 ± 0.73	93.16 ± 0.18	**57.17 ± 0.97**	**93.39 ± 0.38**	**61.81 ± 2.30**	**67.24 ± 3.93**	74.80 ± 0.22	37.82 ± 0.17	**71.95**
Proposed	0.1	82.12 ± 1.37	**79.85 ± 0.33**	**93.22 ± 0.16**	56.90 ± 0.12	93.22 ± 1.10	60.67 ± 0.00	**67.24 ± 0.86**	**76.30 ± 0.37**	37.82 ± 0.02	71.93
Proposed	0.2	**82.77 ± 0.85**	79.72 ± 0.48	93.15 ± 0.31	56.75 ± 0.14	93.38 ± 0.09	61.21 ± 1.64	**67.24 ± 3.09**	74.23 ± 0.39	37.68 ± 0.29	71.79
Proposed	0.5	82.37 ± 1.56	78.99 ± 0.50	92.96 ± 0.24	55.86 ± 0.03	92.78 ± 0.79	60.65 ± 0.33	66.90 ± 1.48	74.76 ± 0.26	38.17 ± 0.91	71.49
Proposed	0.8	80.80 ± 2.00	78.62 ± 0.12	92.82 ± 0.23	55.19 ± 0.15	92.10 ± 0.45	61.36 ± 1.45	66.78 ± 1.98	74.25 ± 0.29	**38.72 ± 0.92**	71.18

tations using an audio retrieval task with auxiliary information [5], one of the practical tasks utilizing semantic differences.

Experimental setup
This experiment used the APwD-Dataset [5], which consists of a set of two similar audio clips and an auxiliary text describing the differences between these audio clips. The task is to search for a target audio that best matches the query audio and auxiliary text. The audio clip is a mixture of ESC-50 audio event samples (foreground sound with class labels) and an FSD50K acoustic scene sample (background sound). This dataset contains two scenes, "Rain" and "Traffic," distinguished by their background sounds, consisting of 50,000/1,000 samples for training and testing sets. In addition, class labels are available for an extra classification task.

We followed [5] for the system and the training/test details. Fig. 3 shows the system that inputs a query audio and a query-modifier text (auxiliary information), and searches the target audio using cosine similarity. During training, it learned through contrastive learning and multi-label classification tasks. We used the encoder pre-trained by the proposed method as the audio embedding layer in the shared audio encoder blocks and DistilBERT [15] as text embedding layer in the text encoder block. We froze all audio/text encoder parameters. We trained the adapter and linear layers for 300 epochs using the Adam [11] optimizer. We assigned 10% of the training samples for validation, and the model with the smallest validation loss was used for evaluation. We used recall@K(R@K) to evaluate the accuracy of audio retrieval. R@K is the rate at which the ground-truth audio files are included in the Kth rank of the selected candidates. We ran the evaluation with three random seeds and averaged the results to obtain the final score.

Results
Table 2 shows that the audio encoder pre-trained by the proposed method improves the audio retrieval performance. The results contain the conventional method [5] using VGGish [16] as audio embedding, the baseline using BYOL-A, and the proposed methods. The "Rain" results show that the proposed method improved to 53.99% for R@1 from the baseline of 50.23% and the conventional 44.5%. The "Traffic" results also show that the proposed method improved to 39.73% for R@1 from the baseline of 36.86% and the conventional 39.1%. These results validate the effectiveness of the proposed subtraction-consistent representation learning for the difference-aware audio task.

5.4. Evaluation: General-purpose audio representation

We validated that the proposed subtraction-consistent representation learning maintains a general-purpose audio representation performance without the impact of learning the difference-aware ability. We followed BYOL-A [3] to assess the performance in a linear evaluation on various tasks, including environmental sound, music, and speech.

Experimental setup
The tasks for linear evaluation include ESC-50 [12], Urban Sound 8K [17] (US8K), Speech Command V2 [18] (SPCV2), VoxCeleb1 [19] (VC1), VoxForge [20] (VF), CREMA-D [21] (CRM-D), GTZAN [22], NSynth [23], and the Pitch Audio Dataset (Surge synthesizer) [24] (Surge). The training/test details follow BYOL-A [3], such as the training epochs 200 with early stopping based on the validation loss. We ran the evaluation with three random seeds and averaged the results with 95% CI.

Results
Table 3 shows that the proposed method slightly degrades the general-purpose performance, while most results are within the 95% confidence interval. The average result degrades from 71.95% for the baseline to 71.18% for $\lambda = 0.8$. However, most task results of the proposed method are marked with underline, i.e., within the range of 95% confidence interval of the baseline results. The most significant degradation of VC1 is -1.98 from 57.17%, which should be a slight drop considering the confidence interval range is ± 0.97. These results confirm that the performance degradation caused by the proposed subtraction-consistent representation learning is generally insignificant.

We confirm that the large λ changes the characteristics of the learned representations as the APwD-Dataset results in Section 5.3. While using $\lambda = 0.8$ degrades the general-purpose performance most in Table 3, using $\lambda = 0.5$ or 0.8 improves the Traffic performance of APwD-Dataset in Table 2. In addition, Surge, a pitch classification of musical instruments, improves as λ becomes larger, suggesting the representation contains more pitch information. These observations suggest a tradeoff of task performance by the use of learning tasks.

6. CONCLUSION

This study investigates how to learn difference-aware audio representations. We propose a self-supervised learning method called subtraction-consistent representation learning. With the obtained representation, mixed sounds are separable by subtracting representations in latent space. In the proposed method, an additional network extending BYOL-A learns the difference between a sound sample and its down-mix with another sound sample. Experiments confirmed that the proposed method improves the accuracy of audio signal retrieval with text auxiliary information utilizing semantic differences in sounds. It was also confirmed that the performance of the proposed method does not degrade significantly in the linear evaluation of various traditional audio classification tasks that require general-purpose audio representation.

7. ACKNOWLEDGEMENT

This work partially supported by JST Strategic International Collaborative Research Program (SICORP), Grant Number JPMJSC2306, Japan.

8. REFERENCES

[1] A. Saeed, D. Grangier, and N. Zeghidour, "Contrastive learning of general-purpose audio representations," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2021, pp. 3875–3879.

[2] E. Fonseca, D. Ortego, K. McGuinness, N. E. O'Connor, and X. Serra, "Unsupervised contrastive learning of sound event representations," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2021, pp. 371–375.

[3] D. Niizumi, D. Takeuchi, Y. Ohishi, N. Harada, and K. Kashino, "BYOL for Audio: Exploring Pre-trained General-purpose Audio Representations," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 31, p. 137–151, 2023.

[4] S. Ghosh, A. Seth, and S. Umesh, "Decorrelating feature spaces for learning general-purpose audio representations," *IEEE J. Sel. Topics Signal Process.*, vol. 16, no. 6, pp. 1402–1414, 2022.

[5] D. Takeuchi, Y. Ohishi, D. Niizumi, N. Harada, and K. Kashino, "Introducing auxiliary text query-modifier to content-based audio retrieval," in *Proc. Interspeech*, 2022.

[6] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko, "Bootstrap your own latent - a new approach to self-supervised learning," in *Proc. Conf. Workshop Neural Inf. Process. Syst. (NeurIPS)*, 2020.

[7] S. Tsubaki, Y. Kawaguchi, T. Nishida, K. Imoto, Y. Okamoto, K. Dohi, and T. Endo, "Audio-change captioning to explain machine-sound anomalies," in *Proc. Detect. Classif. Acoust. Scenes Events (DCASE) Workshop*, 2023, pp. 201–205.

[8] D. Takeuchi, Y. Ohishi, D. Niizumi, N. Harada, and K. Kashino, "Audio difference captioning utilizing similarity-discrepancy disentanglement," in *Proc. Detect. Classif. Acoust. Scenes Events (DCASE) Workshop*, 2023, pp. 181–185.

[9] T. Komatsu, Y. Fujita, K. Takeda, and T. Toda, "Audio difference learning for audio captioning," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2024.

[10] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio Set: An ontology and human-labeled dataset for audio events," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2017, pp. 776–780.

[11] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015.

[12] K. J. Piczak, "ESC: Dataset for Environmental Sound Classification," in *Proc. ACM Multimed.*, 2015, pp. 1015–1018.

[13] J. Salamon, D. MacConnell, M. Cartwright, P. Li, and J. P. Bello, "Scaper: A library for soundscape synthesis and augmentation," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust. (WASPAA)*, 2017, pp. 344–348.

[14] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.

[15] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *arXiv preprint arXiv::1910.01108*, vol. abs/1910.01108, 2019.

[16] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. Weiss, and K. Wilson, "Cnn architectures for largescale audio classification," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2017, pp. 131–135.

[17] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *Proc. ACM Multimed.*, 2014, pp. 1041–1044.

[18] P. Warden, "Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition," *arXiv preprint arXiv::1804.03209*, 2018.

[19] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: A large-scale speaker identification dataset," in *Proc. Interspeech*, 2017, pp. 2616–2620.

[20] K. MacLean, *"Voxforge"*, 2018, available at http://www.voxforge.org/home.

[21] H. Cao, D. G. Cooper, M. K. Keutmann, R. C. Gur, A. Nenkova, and R. Verma, "CREMA-D: Crowd-sourced emotional multimodal actors dataset," *IEEE Trans. Affective Comput.*, vol. 5, no. 4, 2014.

[22] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. Speech Audio Process.*, vol. 10, no. 5, 2002.

[23] J. Engel, C. Resnick, A. Roberts, S. Dieleman, M. Norouzi, D. Eck, and K. Simonyan, "Neural audio synthesis of musical notes with WaveNet autoencoders," in *ICML*, 2017.

[24] J. Turian, J. Shier, G. Tzanetakis, K. McNally, and M. Henry, "One billion audio sounds from GPU-enabled modular synthesis," in *Proc. Int. Conf. Digit. Audio Eff. (DAFx)*, 2021.

THE LANGUAGE OF SOUND SEARCH: EXAMINING USER QUERIES IN AUDIO SEARCH ENGINES

Benno Weck[1,2], *Frederic Font*[2]

[1] Huawei Technologies, Munich Research Center, Germany
[2] Universitat Pompeu Fabra, Music Technology Group, Spain
benno.weck01@estudiant.upf.edu, frederic.font@upf.edu

ABSTRACT

This study examines textual, user-written search queries within the context of sound search engines, encompassing various applications such as foley, sound effects, and general audio retrieval. Current research inadequately addresses real-world user needs and behaviours in designing text-based audio retrieval systems. To bridge this gap, we analysed search queries from two sources: a custom survey and Freesound website query logs. The survey was designed to collect queries for an unrestricted, hypothetical sound search engine, resulting in a dataset that captures user intentions without the constraints of existing systems. This dataset is also made available for sharing with the research community. In contrast, the Freesound query logs encompass approximately 9 million search requests, providing a comprehensive view of real-world usage patterns. Our findings indicate that survey queries are generally longer than Freesound queries, suggesting users prefer detailed queries when not limited by system constraints. Both datasets predominantly feature keyword-based queries, with few survey participants using full sentences. Key factors influencing survey queries include the primary sound source, intended usage, perceived location, and the number of sound sources. These insights are crucial for developing user-centred, effective text-based audio retrieval systems, enhancing our understanding of user behaviour in sound search contexts.

Index Terms— query log analysis, sound search, text-to-audio retrieval, Freesound

1. INTRODUCTION

Users search for foley, sound effects, and other audio elements daily, playing a crucial role in multimedia production, gaming, filmmaking, and various other creative industries. As the demand for high-quality and diverse sound assets grows, understanding user search behaviour becomes increasingly vital for developing efficient and intuitive sound search engines. Platforms like Freesound [1] and FindSounds.com [2] offer robust search functionalities to cater to this growing need for sound resources.

Unlike information retrieval involving purely textual data, multimedia retrieval — and thus sound search — is faced with the problem of a modality gap. To overcome it, different forms of content-based retrieval have been proposed, such as querying by acoustic features or query-by-example [3]. However, these methods are still not widely adopted and most search interfaces on the internet operate primarily with text-based search queries as input. Despite their widespread use, there is a significant gap in research addressing how users formulate search queries on sound search platforms. While previous studies have examined search queries for insights on semantic attributes of sounds [4], no research, to the best of our knowledge, has systematically investigated the nature and characteristics of sound search queries, leaving a critical aspect of user behaviour unexplored.

Examining text queries is particularly valuable given the recent advancements in large language models (LLMs) [5], which have significantly enhanced the feasibility of processing complex natural language inputs across various applications. Furthermore, there is a notable trend towards multi-modal retrieval techniques, which often operate on long-form input texts [6, 7, 8]. Recently a new family of audio retrieval systems focusing on cross-modal retrieval techniques have been proposed [9, 10, 11]. These systems promise to retrieve audio recordings based on text queries by directly matching the text with the audio content. This approach eliminates the need for textual metadata and potentially offers users greater expressive power. However, real-world user needs and behaviours are often overlooked. For example, these text-to-audio retrieval systems typically train on full-form sentence descriptions, whereas actual user inputs may not match this format. As seen in generative systems for automatic music or image generation, user prompts tend to be short and underspecified or, more generally, be out-of-distribution in comparison to the training data [12, 13]. This discrepancy can hinder system performance.

Prior research in web-search and information retrieval shows that people tend to search with short queries [14]. However, expectations towards systems might have shifted due to the widespread adoption of LLMs, and users might provide more text than before. This leaves us to wonder if there is a need to investigate where on the spectrum of input length and complexity user preference falls. This study aims to answer two questions:

RQ1 How would users like to search for sounds using text-only systems?

RQ2 How do people currently use text queries in a real-world sound search system?

In short, the contribution of our work is to shed light on user behaviour, expectations, and the status quo in sound search to guide the development of future sound search systems. We do so by analysing actual search queries from both a custom survey and the Freesound website query logs. Additionally, we provide insights that are important for guiding the development of user-centred, effective text-based audio retrieval systems.

2. METHOD

In an effort to answer our research questions, we collect data from two sources: an online survey and search query logs from the sound-sharing platform Freesound. The survey features a mock search

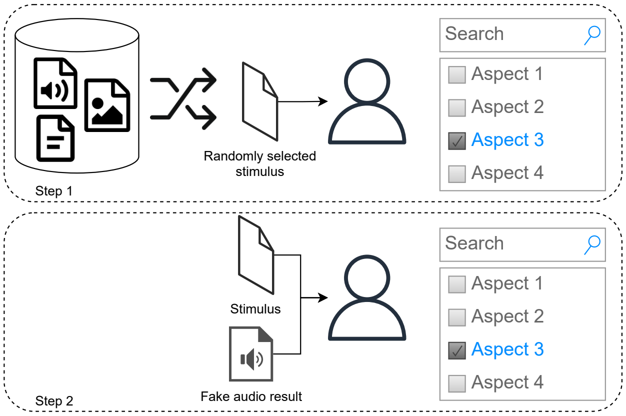


Figure 1: Schematic of the survey workflow: A participant is prompted with a randomly chosen stimulus, asked to provide a search query and indicate aspects that influenced their query. In a subsequent step, a simulated search result is presented together with the stimulus to elicit an updated query.

task designed to elicit queries that allow interpretations of user expectations towards a hypothetical sound search system backed by a limitless retrieval engine. The query log data is selected to reflect the real-world usage of a sound search service. We publish the data collected in the survey online.[1]

2.1. Online survey

We devise a survey to collect user-written text queries with two goals in mind: i) How would users formulate queries if they do not feel restricted by the requirements of a specific search system and ii) what aspects of sounds influence the query formulation.

To ensure realistic and diverse queries, participants were assigned a search task where they could submit and potentially update their queries. The task involved an initial stimulus in the first step and a hypothetical search result in the second step, as outlined in Figure 1. This setup was designed to engage participants while simulating the essential mechanics of a search engine, with the stimulus serving to define a target sound through various modalities. The stimuli, which were randomly assigned, were presented as either a sound recording, an image, or a text description of a sound. In both steps, we additionally ask users what they considered important when writing or refining their query, respectively. More specifically, they select from a list of 12 predefined aspects all that they consider relevant. We list the aspects with a short explanation in Table 1.

2.1.1. Data sources

To not be limited by the performance of an actual retrieval system and to give as much creative freedom to participants during the experiment, we do not employ any actual search engine. Rather, we simulate retrieval results by manually mapping stimuli to audio clips prior to the experiment. The audio clips should serve as examples of results that are somewhat relevant but not fully satisfactory and could require refinement of the query. Moreover, the stimuli are all collected manually to relate to a wide range of potential recordings ranging from natural sounds over instrument samples to sound effects. Specifically, we consider three different types of stimuli:

Aspect	Explanatory description
Main sound source	The most prominent and recognisable object, entity or event in the sound.
Number of sources	How many sound sources there are.
Usage context	What the sound could be used for, e.g. in a movie, in a game, in a commercial
Loudness	How loud or quiet the sound is.
Perceived emotion	How the sound makes you feel.
Recording Quality	The perceived fidelity of the sound, i.e. how clear or noisy it is.
Rhythm	The perceived regularity or irregularity of the sound, e.g. repetitive/chaotic, fast/slow.
Duration	How long the sound lasts, e.g. short/long.
Color and/or density	The perceived quality and/or composition of the sound, e.g. bright/dark, warm/cold, harsh/smooth, simple/complex, etc.
Pitch	The perceived frequency of the sound.
Temporal order	The order in which events occur in time, e.g. first/last, before/after, simultaneously.
Recording setting	The perceived space and environment in which the sound was recorded.

Table 1: Aspect options available to survey respondents.

Audio recordings The FSD50K dataset [15] is chosen as a data source for our audio recording stimuli. It features annotations for 200 sound classes and for each class, a list of example sounds is given. To obtain a stimulus-result pair, a random class is chosen and two distinct sounds of the examples are selected at random.

Images To acquire a set of images that can be linked to matching sounds, we first select 100 sounds from Freesound to be representative of the prevailing sound categories on the platform. For each sound, we search for potential fitting images on the Creative Commons image platform Openverse[2] and select several if possible. Through this curation process, we collect 334 image-sound pairs.

Text descriptions We source the text descriptions from the audio captioning dataset Clotho [16]. More specifically, we synthesise summarative statements from the five crowd-sourced captions belonging to a single sound using the LLM Mixtral 8x7b [17]. Since the associated sound is a perfect match to the description, it can not be used as the search result. Instead, we turn to the TAU Audio-Text Graded Relevance 2023 dataset to find sounds that are relevant to the descriptions [18].

2.1.2. Participation and Participant welfare

To find people interested in sound search, participants were recruited through an announcement on the Freesound website. Participation was completely voluntarily and no compensation was given. During the experiment, participants were free to skip a certain stimulus. Additionally, they are offered to end their participation after

[1] https://doi.org/10.5281/zenodo.13622537

[2] https://openverse.org

completing nine search tasks and every three tasks after that. To not bias our data through high number of annotations by individual participants, the maximum number of search tasks is 21.

Prior to participation, the survey experiment was approved by an Institutional Review Board of the Universitat Pompeu Fabra to ensure alignment with ethical guidelines and protections for human subjects in research. The survey was fully anonymous and did not collect any personal data, safeguarding respondents' privacy and confidentiality. Participants were informed about the objectives of the research, their tasks, and the use of their survey answers, underpinning their informed consent before contributing to the project.

2.2. Query log analysis

In addition to the survey, we collect anonymised system logs for search queries conducted on the Freesound website. The text search on Freesound matches the textual metadata (user-provided sound titles and descriptions) and allows users to filter results according to various aspects including file type, sampling rate, etc. Since our focus is on textual queries, we exclude all requests that do not specify a query or rely on search filters. We consider search requests submitted over the course of 12 weeks from April to June 2024 and collected a total of 9 M queries. Table 2 outlines the structure of the query log data.

For further data analysis, we apply a series of processing steps. First, all queries are case-folded. Then, to detect search requests that were submitted by a single user in a sequential fashion, i.e. likely belonging to the same session, we group on the timestamp and anonymised IP address. Adopting a popular baseline method in session detection, we assign requests to distinct sessions if they are separated by at least 30 minutes [19]. Finally, to better understand what people are searching for, we take all queries submitted by at least 100 different IPs and manually annotate them with a single topic. The list of topics for annotation was adopted from the AudioSet taxonomy [20]. If a query term is ambiguous (e.g. 'metal', 'swing' or 'kick') it is left unannotated. All annotations were done by one annotator. In total, we could annotate 978 of the 1,000 most common search queries and we share the annotations in the same repository as the survey results (see Sec. 2.1).

timestamp	anonymised IP addr.	query
20240603073000	6ff843ba...	"dog"
20240603073050	6ff843ba...	"dog barking"
20240603073150	d24f26cf...	"background music"

Table 2: Excerpt of query logs collected from Freesound.

3. RESULTS

3.1. Survey results

In our survey, 94 participants completed a total of 706 search tasks with an average of 7.5 (median 9.0) tasks per participant. The mean time spent on a single task is 97.8 seconds. All three stimuli types are approximately equally represented in the data with 240 data points for image stimuli, 238 for audio, and 228 for text, respectively. The initial query contained 4.4 tokens on average and 5.5 tokens after refinement in the second step of the search task. Queries were slightly longer when based on a text stimulus (median: four tokens) in comparison with the other two types of stimuli (median: three tokens). Participants chose to not update their query in the second step, or submitted the same query verbatim, in 36% of cases.

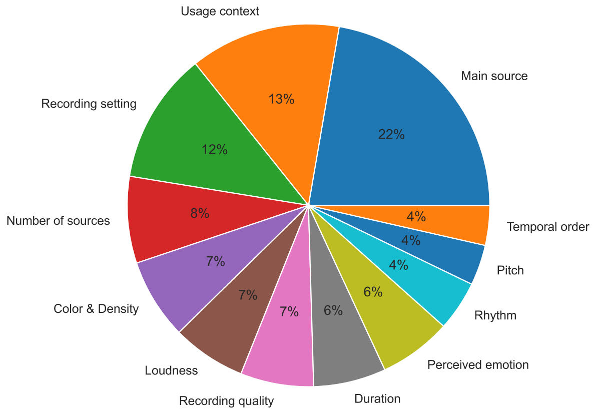


Figure 2: Distribution of aspects chosen by survey respondents to indicate what they considered important when searching for sounds.

When reviewing how participants chose to update their query, we find that most commonly the updated queries are longer by one token (32% of the cases) or two tokens (17%), but sometimes also keep the same number of tokens (14%) or are shorter by at least one token (18%). Examples of these updates include: 'water dripping' → 'slow water dripping', 'car passing by' → 'car passing by in distance', and 'Creaking Door' → 'Creaking Door Opening and Closing'. Overall, queries consist of enumerated keywords (e.g. 'drums, instrumental, live sound, music', 'fishermen dock crowd'), short noun or verb phrases ('lively restaurant room', 'percussion instruments', 'child playing toy harmonica') or a combination thereof ('short clip of rapid intake of breath, moderately high pitch'). Only very few participants formed full sentences ('Man gives great speech', 'Water is flowing through pipes'). Negations are rare and mostly only present in the refined queries (e.g. 'live guitar' → 'live guitar no synth').

From the Figure 2, we can see that participants consider aspects relating to the content of the sound (main sound source, number of sources & recording setting) most important. Moreover, the data indicates that the usage context of a sound influences users search behaviour. Upon closer inspection of the query terms however, we can not find this reflected in the queries, i.e. there are hardly any words that describe a usage context. Finally, aspects related to perceptual properties (Loudness, Colour and density, Pitch) and structural attributes (Duration, Rhythm, Temporal order) of the audio recording were given less attention.

3.2. Query log analysis

The mean query length in the query log data is 1.8 (median: 2) and the average number of queries per session is 3.9 (median: 2). In contrast to the survey experiment, we cannot find a significant increase in query length for subsequent requests within a session.

Figure 3 shows the breakdown of the topics found when annotating the query log data with the first two levels of the AudioSet ontology. We extend the ontology with a new category ('Other')

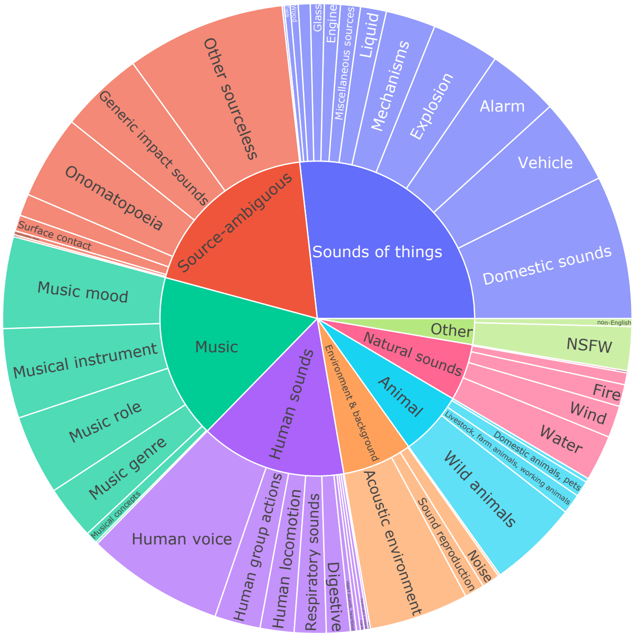


Figure 3: Distribution of topics found in the top 1,000 search queries submitted on Freesound.

Category	Search query examples
utterances and vocables	‘oh no’, ‘yeah’, ‘hmm’, ‘yay’, ‘huh’, ‘hello’, ‘hey’, ‘what’
production jargon	‘riser, ‘one shot’, ‘stab’, ‘stinger’, ‘bumper’
abbreviations	‘atmo’, ‘bgm’
intended use	‘error’, ‘success’, ‘correct answer’, ‘alert’, ‘button click’, ‘game over’, ‘jumpscare’

Table 3: Examples of special vocabulary used in sound search.

including non-English texts and queries related to NSFW content. What stands out in this chart is that queries are generally related to a wide range of topics and span across all classes of the taxonomy. There is a high interest in sound effects, recordings relating to objects, music, and human-made sounds.

Moreover, as a side-effect of manually annotating the topics, we identify interesting patterns in the expressions used in the queries that highlight another dimension of user search behaviour. Table 3 lists broad categories for these expressions ranging from jargon specific to sound design, music and video production, etc. to literal use of single words to find short speech recordings.

4. DISCUSSION AND LIMITATIONS

The above-presented results indicate that users generally tend to use short queries when using sound search systems and that there is no expectation from users that complex queries such as describing interactions between elements or temporal order are understood or helpful to achieve their search goal. While these results are expected in light of the findings in the literature on search behaviour, the most striking difference in our study is that queries collected in the survey were significantly longer than those of the query log. These data must be interpreted with caution since the specifics of the Freesound search system might encourage users to submit short queries. By default, all search terms provided in a user query must be present in a document to match the query, i.e. for a sound to be a returned result to the query “dog barking baby crying” all four words must be found in the metadata. Nonetheless, it leads us to the hypothesis that users of sound search systems would provide longer and potentially more complex queries if the system supports it.

Reflecting on the latest developments in text-to-audio retrieval research, our analysis shows a discrepancy between the existing datasets and potential user input. These datasets are usually repurposed from the task of audio captioning and we argue that they are inadequate for two main reasons. Firstly, datasets commonly used for evaluation and benchmarking such as the Clotho dataset might not give a reliable estimate of real-world performance due to the choice of audio recordings. For example, the creators of Clotho purposely exclude music, sound effects, and speech recordings [16], while it is apparent from our analysis that user interest is spread over a wide range of topics. Secondly, the way people formulate their queries might present a challenge, as user input often takes the form of short enumerations or keywords rather than full sentences, which contrasts with the textual training and evaluation data that typically consist of complete sentences.

The generalisability of the presented results is subject to certain limitations. For instance, one limitation of our study lies in the fact that recruitment for the experimental survey was done via the Freesound website. Responses might be biased by participants’ experience with the website’s search engine. Furthermore, the results regarding the importance of aspects in the experimental survey provide a limited view of the participants’ motivations since they are heavily influenced by the choice of stimulus. Finally, the relatively small sample size in topic annotations limits the comprehensiveness of our findings, highlighting the need for future research to expand and deepen our understanding of user behaviour and preferences.

Further research might also explore users’ intentions in sound search since we mostly provide a view on the “what” dimension and not the “why” of search [21]. We see from both sets of results that searchers (not surprisingly) focus on the main elements comprising a sound in their queries. These results are in agreement with the observations of Giordano et al. [22], who suggested that “the most informative way to describe natural sounds verbally focuses on the properties of the sound source, rather than on sensory or acoustic attributes.” However, understanding the underlying intentions is necessary to ultimately improve search performance satisfaction.

5. CONCLUSIONS AND FUTURE WORK

Our study analysed sound search queries from two sources: submitted by participants of an online survey and the query log of Freesound. The results of this investigation suggest that users of sound search systems would provide longer queries if not limited by system constraints. The second major finding was a clear discrepancy between user-written queries and current research datasets in text-to-audio retrieval research with respect to the topics covered and the language used. Future work should look into creating datasets specifically designed for the purpose of evaluating sound retrieval systems with user expectations and behaviour in mind.

6. REFERENCES

[1] F. Font, G. Roma, and X. Serra, “Freesound Technical Demo,” in *Proceedings of the 21st ACM International Conference on Multimedia*, ser. MM ’13. Barcelona, Spain: Association for Computing Machinery, New York, NY, United States, 2013, pp. 411–412. [Online]. Available: https://doi.org/10.1145/2502081.2502245

[2] S. Rice and S. Bailey, “A Web Search Engine for Sound Effects,” *Journal of the Audio Engineering Society*, Oct. 2005. [Online]. Available: https://aes2.org/publications/elibrary-page/?id=13281

[3] B. Kim and B. Pardo, “Improving Content-based Audio Retrieval by Vocal Imitation Feedback,” in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Brighton, United Kingdom: IEEE, May 2019, pp. 4100–4104. [Online]. Available: https://ieeexplore.ieee.org/document/8683461/

[4] A. Pearce, T. Brookes, and R. Mason, “Timbral Attributes for Sound Effect Library Searching,” in *AES International Conference Semantic Audio 2017, Erlangen, Germany, June 22-24, 2017*, C. Dittmar, J. Abeßer, and M. Müller, Eds. Audio Engineering Society, 2017. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=18754

[5] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, and J. Gao, “Large Language Models: A Survey,” Feb. 2024, arXiv:2402.06196 [cs]. [Online]. Available: http://arxiv.org/abs/2402.06196

[6] K. Zhou, F. H. Hassan, and G. K. Hoon, “The State of the Art for Cross-Modal Retrieval: A Survey,” *IEEE Access*, vol. 11, pp. 138 568–138 589, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10336787/

[7] P. Kaur, H. S. Pannu, and A. K. Malhi, “Comparative analysis on cross-modal information retrieval: A review,” *Computer Science Review*, vol. 39, p. 100336, Feb. 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1574013720304366

[8] M. Levy, R. Ben-Ari, N. Darshan, and D. Lischinski, “Chatting Makes Perfect: Chat-based Image Retrieval,” in *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., vol. 36. Curran Associates, Inc., 2023, pp. 61 437–61 449.

[9] A. S. Koepke, A.-M. Oncescu, J. Henriques, Z. Akata, and S. Albanie, “Audio Retrieval with Natural Language Queries: A Benchmark Study,” *IEEE Transactions on Multimedia*, pp. 1–1, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9707629/

[10] H. Xie, S. Lipping, and T. Virtanen, “Language-Based Audio Retrieval Task in DCASE 2022 Challenge,” in *Proceedings of the 7th Workshop on Detection and Classification of Acoustic Scenes and Events 2022, DCASE 2022, Nancy, France*, M. Lagrange, A. Mesaros, T. Pellegrini, G. Richard, R. Serizel, and D. Stowell, Eds., Nov. 2022. [Online]. Available: https://dcase.community/documents/workshop2022/proceedings/DCASE2022Workshop_Xie_56.pdf

[11] B. Elizalde, S. Zarar, and B. Raj, “Cross Modal Audio Search and Retrieval with Joint Embeddings Based on Text and Audio,” in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Brighton, United Kingdom: IEEE, May 2019, pp. 4095–4099. [Online]. Available: https://ieeexplore.ieee.org/document/8682632/

[12] E. Chang, S. Srinivasan, M. Luthra, P.-J. Lin, V. Nagaraja, F. Iandola, Z. Liu, Z. Ni, C. Zhao, Y. Shi, and V. Chandra, “On The Open Prompt Challenge In Conditional Audio Generation,” Nov. 2023, arXiv:2311.00897 [cs, eess]. [Online]. Available: http://arxiv.org/abs/2311.00897

[13] Y. Xie, Z. Pan, J. Ma, L. Jie, and Q. Mei, “A Prompt Log Analysis of Text-to-Image Generation Systems,” in *Proceedings of the ACM Web Conference 2023*. Austin TX USA: ACM, Apr. 2023, pp. 3892–3902. [Online]. Available: https://dl.acm.org/doi/10.1145/3543507.3587430

[14] R. Baeza-Yates, C. Hurtado, M. Mendoza, and G. Dupret, “Modeling User Search Behavior,” in *Third Latin American Web Congress (LA-WEB’2005)*. Buenos Aires, Argentina: IEEE, 2005, pp. 242–251. [Online]. Available: http://ieeexplore.ieee.org/document/1592383/

[15] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, “FSD50K: an open dataset of human-labeled sound events,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 829–852, 2022.

[16] K. Drossos, S. Lipping, and T. Virtanen, “Clotho: an Audio Captioning Dataset,” in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Barcelona, Spain: IEEE, May 2020, pp. 736–740. [Online]. Available: https://ieeexplore.ieee.org/document/9052990/

[17] A. Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D. S. Chaplot, *et al.*, “Mixtral of Experts,” 2024, version Number: 1. [Online]. Available: https://arxiv.org/abs/2401.04088

[18] H. Xie, K. Khorrami, O. Räsänen, and T. Virtanen, “Crowdsourcing and Evaluating Text-Based Audio Retrieval Relevances,” in *Proceedings of the 8th Detection and Classification of Acoustic Scenes and Events 2023 Workshop (DCASE2023)*, Tampere, Finland, Sept. 2023, pp. 226–230.

[19] D. Gayo-Avello, “A survey on session detection methods in query logs and a proposal for future evaluation,” *Information Sciences*, vol. 179, no. 12, pp. 1822–1843, May 2009. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S002002550900053X

[20] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, “Audio Set: An ontology and human-labeled dataset for audio events,” in *Proc. IEEE ICASSP 2017*, New Orleans, LA, 2017.

[21] C. Kofler, M. Larson, and A. Hanjalic, “User Intent in Multimedia Search: A Survey of the State of the Art and Future Challenges,” *ACM Computing Surveys*, vol. 49, no. 2, pp. 1–37, June 2017. [Online]. Available: https://dl.acm.org/doi/10.1145/2954930

[22] B. L. Giordano, R. De Miranda Azevedo, Y. Plasencia-Calaña, E. Formisano, and M. Dumontier, “What do we mean with sound semantics, exactly? A survey of taxonomies and ontologies of everyday sounds,” *Frontiers in Psychology*, vol. 13, p. 964209, Sept. 2022. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fpsyg.2022.964209

ADAPROJ: ADAPTIVELY SCALED ANGULAR MARGIN SUBSPACE PROJECTIONS FOR ANOMALOUS SOUND DETECTION WITH AUXILIARY CLASSIFICATION TASKS

Kevin Wilkinghoff

Fraunhofer FKIE, Fraunhoferstr. 20 53353 Wachtberg, Germany
kevin.wilkinghoff@ieee.org

ABSTRACT

The state-of-the-art approach for semi-supervised anomalous sound detection is to first learn an embedding space by using auxiliary classification tasks based on meta information or self-supervised learning and then estimate the distribution of normal data. In this work, AdaProj a novel loss function for training the embedding model is presented. In contrast to commonly used angular margin losses, which project data of each class as close as possible to their corresponding class centers, AdaProj learns to project data onto class-specific subspaces while still ensuring an angular margin between classes. By doing so, the resulting distributions of the embeddings belonging to normal data are not required to be as restrictive as other loss functions allowing a more detailed view on the data. In experiments conducted on the DCASE2022 and DCASE2023 anomalous sound detection datasets, it is shown that using AdaProj to learn an embedding space significantly outperforms other commonly used loss functions.

Index Terms— machine listening, anomaly detection, representation learning, domain generalization

1. INTRODUCTION

Semi-supervised anomaly detection is the task of training a system to differentiate between normal and anomalous data using only normal training samples [1]. An example application is acoustic machine condition monitoring for predictive maintenance [2, 3]. Here, normal data corresponds to sounds of fully functioning machines whereas anomalous sounds indicate mechanical failure. One of the main difficulties to overcome in acoustic machine condition monitoring is that it is practically impossible to record isolated sounds of a target machine. Instead, recordings also contain many other sounds emitted by non-target machines or other sound sources such as humans. Compared to this complex acoustic scene, anomalous signal components of the target machines are very subtle and hard to detect without utilizing additional knowledge. Another main difficulty is that a system should also be able to reliably detect anomalous sounds when changing the acoustic conditions or machine settings without needing to collect large amounts of data in these changed conditions or to re-train the system (domain generalization [4]). One possibility to overcome both difficulties is to learn a mapping of the audio signals into a fixed-dimensional vector space, in which representations belonging to normal and anomalous data, called embeddings, can be easily separated. Then, by estimating the distribution of normal training samples in the embedding space, one can compute an anomaly score for a test sample to distinguish between normal and anomalous samples.

To train such an embedding model, the state-of-the-art is to utilize an auxiliary classification task using provided meta information or self-supervised learning. This enables the embedding model to closely monitor target signals and ignore other signals and noise [5]. For machine condition monitoring, possible auxiliary tasks are classifying between machine types [6–8] or, additionally, between different machine states and noise settings [9–11], recognizing augmented and non-augmented versions of normal data [6, 12] or predicting the activity of machines [10]. Using an auxiliary task to learn embeddings is also called outlier exposure (OE) [13] because normal samples belonging to other classes than a target class can be considered proxy outliers [14].

The contributions of this work are the following. First and foremost, AdaProj, a novel angular margin loss function that learns class-specific subspaces for training an embedding model, is presented[1]. Second, it is proven that AdaProj has arbitrary large optimal solution spaces allowing to relax the compactness requirements of the class-specific distributions in the embedding space. Last but not least, AdaProj is compared to other commonly used loss functions. In experiments conducted on the DCASE2022 and DCASE2023 anomalous sound detection (ASD) datasets it is shown that AdaProj outperforms other commonly used loss functions.

1.1. Related Work

When training a neural network to solve a classification task, usually the softmax function in combination with the categorical cross-entropy (CXE) is used. However, this only reduces inter-class similarity without explicitly increasing intra-class similarity [15]. When training an embedding model for anomaly detection, high intra-class similarity is a desired property to cluster normal data and be able to detect anomalous samples. There are several loss functions that explicitly increase intra-class similarity: [16] proposed a compactness loss to project the data into a hypersphere of minimal volume for one-class classification. However, for machine condition monitoring in noisy conditions it is known that one-class losses perform worse than losses that also discriminatively solve an auxiliary classification task [5]. [17] utilized an additional descriptiveness loss consisting of a CXE imposing a classification task on another arbitrary dataset than the target dataset to regularize the training objective. For machine condition monitoring, often meta information is available as it can at least be ensured which machine is being recorded when collecting data. [8] used center loss [18], which minimizes the distance to learned class centers for each class. Another choice are angular margin losses that learn an embedding space on the unit sphere while ensuring a margin between classes, which improves the generalization capabilities. Specific examples are the additive margin softmax loss [15] as used by [7, 19] and ArcFace [20]

[1] An open-source implementation of the AdaProj loss is available at: `https://github.com/wilkinghoff/AdaProj`

as used by [6, 11, 21]. [22, 23] use the AdaCos loss [24], which essentially is ArcFace with an adaptive scale parameter, or the sub-cluster AdaCos loss [25], which utilizes multiple sub-clusters for each class instead of a single one.

As stated before, the goal of this work is to reduce the restrictions on the learned distributions in the embedding space by learning class-specific linear subspaces. There are also other works on losses aiming at learning subspaces based on orthogonal projections in an embedding space. [26] used orthogonal projections as a constraint for training an autoencoder based anomaly detection system. Another example is semi-supervised image classification by using a combination of class-specific subspace projections with a reconstructions loss and ensure that they are different by also using a discriminative loss [27]. Our work focuses on learning an embedding space through an auxiliary classification task that is well-suited for semi-supervised anomaly detection.

2. METHODOLOGY

2.1. Notation

Let $\phi : X \rightarrow \mathbb{R}^D$ denote a neural network where X denotes some input space, which consists of audio signals in this work, and $D \in \mathbb{N}$ denotes the dimension of the embedding space. Define the linear projection of $x \in \mathbb{R}^D$ onto the subspace $\mathrm{span}(\mathcal{C}_k) \subset \mathbb{R}^D$ as $P_{\mathrm{span}(\mathcal{C}_k)}(x) := \sum_{c_k \in \mathcal{C}_k} \langle x, c_k \rangle c_k$. Furthermore, let $\mathcal{S}^{D-1} = \{y \in \mathbb{R}^D : \|y\|_2 = 1\} \subset \mathbb{R}^D$ denote the D-sphere and define $P_{\mathcal{S}^{D-1}}(x) := \frac{x}{\|x\|_2} \in \mathcal{S}^{D-1}$ to be the projection onto the D-sphere.

2.2. AdaProj loss function

Similar to the sub-cluster AdaCos loss [25], the idea of the AdaProj loss is to enlarge the space of optimal solutions to allow the network to learn less restrictive distributions of normal data. This relaxation is achieved by measuring the distance to class-specific subspaces while training the embedding model instead of measuring the distance to a single or multiple centers as done for other angular margin losses and may help to differentiate between normal and anomalous embeddings after training. The reason is that for some auxiliary classes a strong compactness may be detrimental when aiming to learn an embedding space that separates normal and anomalous data since both may be distributed very similarly.

Formally, the definition of the AdaProj loss is as follows.

Definition 1 (AdaProj loss)**.** Let $\mathcal{C}_k \subset \mathbb{R}^D$ with $|\mathcal{C}_k| = J \in \mathbb{N}$ denote class centers for class $k \in \{1, ..., N_{\text{classes}}\}$. Then for the AdaProj loss the logit for class $k \in \{1, ..., N_{\text{classes}}\}$ is defined as

$$L(x, \mathcal{C}_k) := \hat{s} \cdot \|P_{\mathcal{S}^{D-1}}(x) - P_{\mathcal{S}^{D-1}}(P_{\mathrm{span}(\mathcal{C}_k)}(x))\|_2^2$$

where $\hat{s} \in \mathbb{R}_+$ is the adaptive scale parameter of the AdaCos loss [24]. Inserting these logits into a softmax function and computing the CXE yields the AdaProj loss function.

Remark. Note that, by Lemma 5 of [5], it holds that

$$\begin{aligned}&\|P_{\mathcal{S}^{D-1}}(x) - P_{\mathcal{S}^{D-1}}(P_{\mathrm{span}(\mathcal{C}_k)}(x))\|_2^2\\ =&2(1 - \langle P_{\mathcal{S}^{D-1}}(x), P_{\mathcal{S}^{D-1}}(P_{\mathrm{span}(\mathcal{C}_k)}(x))\rangle),\end{aligned}$$

which is equal to the cosine distance in this case and explains why the AdaProj loss can be called an angular margin loss.

As for other angular margin losses, projecting the embedding space onto the D-sphere has several advantages [5]. Most importantly, if D is sufficiently large randomly initialized centers are with very high probability approximately orthonormal to each other [28], i.e. distributed equidistantly, and sufficiently far away from $\mathbf{0} \in \mathbb{R}^D$. Therefore, one does not need to carefully design a method to initialize the centers. Another advantage is that a normalization may prevent numerical issues, similar to applying batch normalization [29].

The following Lemma shows that using the AdaProj loss, as defined above, indeed increases the solution space.

Lemma 2. *Let $x \in \mathbb{R}^D$ and let $\mathcal{C} \subset \mathbb{R}^D$ contain pairwise orthonormal elements. If $x \in \mathrm{span}(\mathcal{C}) \cap \mathcal{S}^{D-1}$, then*

$$\|P_{\mathcal{S}^{D-1}}(x) - P_{\mathcal{S}^{D-1}}(P_{\mathrm{span}(\mathcal{C})}(x))\|_2^2 = 0.$$

Proof. Let $x \in \mathrm{span}(\mathcal{C}) \cap \mathcal{S}^{D-1} \subset \mathbb{R}^D$ with $|\mathcal{C}| = J$. Therefore, $\|x\|_2 = 1$ and there are $\lambda_j \in \mathbb{R}$ with $x = \sum_{j=1}^J \lambda_j c_j$. Thus, it holds that

$$\begin{aligned}x =& \sum_{j=1}^J \lambda_j c_j = \sum_{j=1}^J \sum_{i=1}^J \lambda_i \langle c_i, c_j \rangle c_j = \sum_{j=1}^J \langle \sum_{i=1}^J \lambda_i c_i, c_j \rangle c_j\\ =& \sum_{j=1}^J \langle x, c_j \rangle c_j = P_{\mathrm{span}(\mathcal{C})}(x).\end{aligned}$$

Hence, we obtain

$$\|P_{\mathcal{S}^{D-1}}(x) - P_{\mathcal{S}^{D-1}}(P_{\mathrm{span}(\mathcal{C})}(x))\|_2^2 = \quad 0.$$

□

Remark. If $\mathcal{C}$ contains randomly initialized elements of the unit sphere and D is sufficiently large, then the elements of $\mathcal{C}$ are approximately pairwise orthonormal with very high probability [28].

When inserting the projection onto the $D-1$-sphere as an operation into the neural network, this Lemma shows that the solution space for the AdaProj loss function is increased to the whole subspace $\mathrm{span}(\mathcal{C})$, which has a dimension of $|\mathcal{C}|$ with very high probability. Because of this, it should be ensured that $|\mathcal{C}| < D$. Otherwise the whole embedding space may be an optimal solution and thus the network cannot learn a meaningful embedding space. In comparison, for the AdaCos loss only the class centers themselves are optimal solutions and for the sub-cluster AdaCos loss each sub-cluster is an optimal solution [5].

3. EXPERIMENTAL RESULTS

3.1. Datasets and performance metrics

For the experiments, the DCASE2022 ASD dataset [2] and the DCASE2023 ASD dataset [3] for semi-supervised machine condition monitoring were used. Both datasets consist of a development set and an evaluation set that are divided into a training split containing only normal data and a test split containing normal as well as anomalous data. Furthermore, both tasks explicitly capture the problem of domain generalization [4] by defining a source and a target domain, which differs from the source domain by altering machine parameters or noise conditions. The task is to detect anomalous samples regardless of the domain a sample belongs to by training a system with only normal data. As meta information, the target machine type of each sample is known and for the training samples, also the domain and additional parameter settings or noise conditions, called attribute information, are known and thus can be utilized to train an embedding model.

The DCASE2022 ASD dataset [2] consists of the machine types "ToyCar" and "ToyTrain" from ToyAdmos2 [30] and "fan", "gearbox", "bearing", "slide rail" and "valve" from MIMII-DG [31]. For each machine type, there are six different sections corresponding to different domain shifts and also defining subsets used for computing the performance. These sections are known for each recording and can also be utilized as meta information to train the system. For the source domain of each section, there are 1000 normal audio recordings with a duration of 10 s and a sampling rate of 16 kHz belonging to the training split. For the target domain of each section, there are only 10 normal audio recordings belonging to the training split. The test splits of each section contain approximately 100 normal and 100 anomalous samples.

The DCASE2023 ASD dataset [3] is similar to the DCASE2022 ASD dataset with the following modifications. First of all, the development set and the evaluation set contain mutually exclusive machine types. More concretely, the development set contains the same machine types as the DCASE2022 dataset and the evaluation set contains the machine types "ToyTank", "ToyNscale" and "ToyDrone" from ToyAdmos2+ [32] and "vacuum", "bandsaw", "grinder" and "shaker" from MIMII-DG [31]. Furthermore, there is only a single section for each machine type, which makes the auxiliary classification task much easier resulting in less informative embeddings for the ASD task. Last but not least, the duration of each recording has a length between 6 s and 18 s. Overall, all three modifications make this task much more challenging than the DCASE2022 ASD task.

To measure the performance of the ASD systems the threshold-independent area under the receiver operating characteristic (ROC) curve (AUC) metric is used. In addition, the partial area under the ROC curve (pAUC) [33], which is the AUC for low false positive rates ranging from 0 to p, with $p = 0.1$, is used. Both performance metrics are computed domain-independent for every previously defined section of the dataset and the harmonic mean of all resulting values is the final score used to measure and compare the performances of different ASD systems.

3.2. Anomalous sound detection system

For all experiments conducted in this work, the state-of-the-art ASD system presented in [23] is used. An overview of the system can be found in Figure 1. The system consists of three main components: 1) a feature extractor, 2) an embedding model and 3) a backend for computing anomaly scores.

In the first processing block, two different feature representations are extracted from the raw waveforms, namely magnitude spectrograms and the magnitude spectrum. To capture less similar information with both feature representations, the temporal mean is subtracted from the spectrograms, essentially removing static frequency information that are captured with the highest possible resolution by the spectra.

For each of the two feature representations, another convolutional sub-network is trained and the resulting embeddings are concatenated and normalized with respect to the Euclidean norm to obtain a single embedding. In contrast to the original architecture, the embedding dimension is doubled from 256 to 512 to increase the likelihood of two randomly initialized center vectors to be orthogonal. Note that even if some of the randomly sampled class centers are not orthonal, the probability that they are linearly independent is equal to 1 if $J < D$. Thus, the subspaces spanned be the class centers do not collapse. More details about the subnetwork architectures can be found in [23]. The network is trained for 10 epochs using a batch size of 64 using adam [34] by utilizing meta information such as machine types and the provided attribute information as an auxiliary classification task. Different loss functions can be used for this purpose and will be compared in the next subsection. All loss functions investigated in this work require class-specific center vectors, which are initialized randomly using Glorot uniform initialization [35]. To improve the ASD performance, the class centers are not adapted during training and no bias terms are used as proposed in [16] for one-class classification. Furthermore, mixup [36] with a uniformly distributed mixing coefficient is applied to the waveforms.

As a backend, k-means with 32 means is applied to the normal training samples of the source domain. For a given test sample, the smallest cosine distance to these means and the ten normal training samples of the target domain is used as an anomaly score. Thus, smaller values indicate normal samples whereas higher values indicate anomalous samples.

3.3. Performance evaluation

In the first experiment, the ASD performance obtained with the following loss functions was compared: 1) individual class-specific intra-class (IC) compactness losses jointly trained on all classes [16] 2) an additional discriminative CXE loss, similar to the descriptiveness loss used in [17] but trained on the same dataset, 3) the AdaCos loss [24], 4) the sub-cluster AdaCos loss [25] with 32 sub-clusters and 5) the proposed AdaProj loss. Each experiment was repeated ten times to reduce the variance of the resulting performances. The results can be found in Table 1.

The main observation to be made is that the proposed AdaProj loss outperforms all other losses. Especially on the DCASE2023 dataset, there are significant improvements to be observed. The most likely explanation is that for this dataset the classification task is less difficult and thus a few classes may be easily identified leading to embeddings that do not carry enough information to distinguish between embeddings belonging to normal and anomalous samples of these classes.

Another interesting observation is that, in contrast to the original results presented in [25], the sub-cluster AdaCos loss actually performs slightly worse than the AdaCos loss despite having a higher solution space. A possible explanation is that in [25], the centers are adapted during training whereas, in our work, they are not as this has been shown to improve the resulting performance [23]. Since all centers have approximately the same distance to each other when being randomly initialized, i.e. the centers belonging to a target class and the other centers, the network will likely utilize only a single center for each class that is closest to the initial embeddings of the corresponding target class. Moreover, a low inter-class similarity is more difficult to ensure due to the higher total number of sub-clusters belonging to other classes. This leads to more restrictive requirements when learning class-specific distributions and thus actually reduces the ability to differentiate between embeddings belonging to normal and anomalous samples.

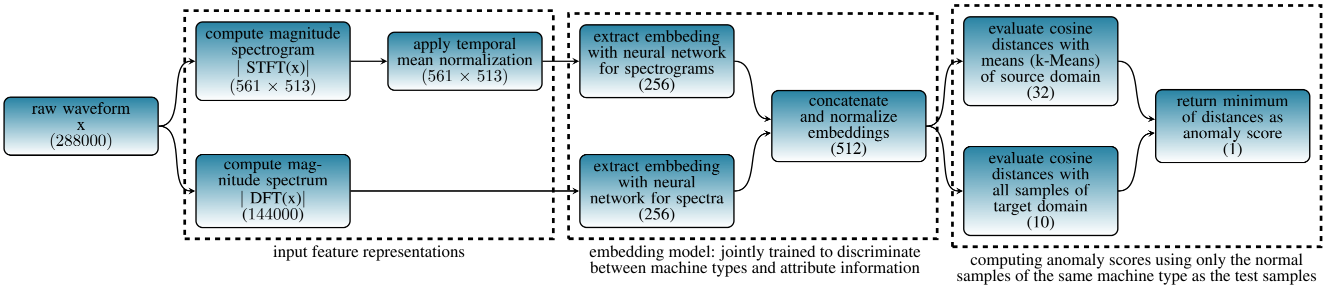

raw waveform
x
(288000)
compute magnitude spectrogram
| STFT(x)|
(561 × 513)
apply temporal mean normalization
(561 × 513)
compute magnitude spectrum
| DFT(x)|
(144000)
input feature representations
extract embbeding with neural network for spectrograms
(256)
extract embedding with neural network for spectra
(256)
concatenate and normalize embeddings
(512)
embedding model: jointly trained to discriminate between machine types and attribute information
evaluate cosine distances with means (k-Means) of source domain
(32)
evaluate cosine distances with all samples of target domain
(10)
return minimum of distances as anomaly score
(1)
computing anomaly scores using only the normal samples of the same machine type as the test samples

Figure 1: Structure of the ASD system, adapted from Figure 1 in [23]. Representation size in each step is given in brackets.

Table 1: ASD performance obtained with different loss functions. Harmonic means of all AUCs and pAUCs over all pre-defined sections of the dataset are depicted in percent. Arithmetic mean and standard deviation of the results over ten independent trials are shown. Best results in each column are highlighted with bold letters.

loss function	DCASE2022 dev. set [2]		DCASE2022 eval. set [2]		DCASE2023 dev. set [3]		DCASE2023 eval. set [3]		arithmetic mean	
	AUC	pAUC	AUC	pAUC	AUC	pAUC	AUC	pAUC	AUC	pAUC
IC compactness loss [16]	79.2 ± 0.9	64.7 ± 1.1	70.3 ± 0.8	58.9 ± 0.8	67.7 ± 1.2	56.9 ± 0.9	64.0 ± 1.5	55.8 ± 0.9	70.3	59.1
IC compactness loss + CXE [17]	79.0 ± 0.8	65.0 ± 0.7	72.6 ± 0.4	60.3 ± 0.7	70.4 ± 1.0	$\mathbf{57.4 \pm 1.1}$	67.5 ± 0.8	57.5 ± 1.0	72.4	60.1
AdaCos loss [24]	79.8 ± 0.7	$\mathbf{65.5 \pm 0.9}$	73.0 ± 0.4	59.7 ± 0.6	70.9 ± 0.9	56.8 ± 0.9	68.0 ± 1.6	58.0 ± 1.1	72.9	60.0
sub-cluster AdaCos loss [25]	80.0 ± 1.4	65.2 ± 1.1	72.9 ± 0.6	59.5 ± 0.5	70.4 ± 0.9	56.3 ± 0.8	66.5 ± 1.6	56.2 ± 1.0	72.5	59.3
proposed AdaProj loss	$\mathbf{80.6 \pm 0.8}$	$\mathbf{65.5 \pm 1.3}$	$\mathbf{73.6 \pm 0.7}$	$\mathbf{60.5 \pm 0.7}$	$\mathbf{71.4 \pm 1.0}$	56.2 ± 0.7	$\mathbf{69.8 \pm 1.3}$	$\mathbf{60.0 \pm 0.5}$	**73.9**	**60.6**

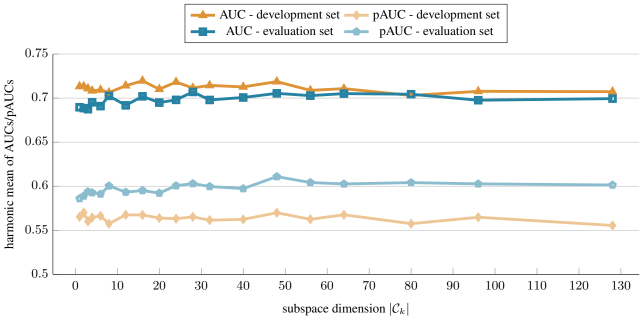


Figure 2: Domain-independent performance obtained on the DCASE2023 dataset with different subspace dimensions. The means over ten independent trials are shown.

3.4. Investigating the impact of the subspace dimension on the performance

As an ablation study, different choices for the dimension of the subspaces have been compared experimentally on the DCASE2023 ASD dataset. The results can be found in Figure 2. It can be seen, that, on the development set, the results are relatively stable while a larger dimension slightly improves the performance on the evaluation set without any significant differences. For subspace dimensions greater than 48 the performances seem to slightly degrade again. In conclusion, the subspace dimension should be neither too high nor too low but other than that appears to not have a significant impact on the performance. Thus, a dimension of 32, as used for the other experiments in this work, appears to be a reasonable choice. Since using the AdaProj with this subspace dimension also outperformed the other loss functions on the DCASE2022 ASD dataset (cf. Table 1, this particular dimension may serve as a default hyperparameter setting for the AdaProj loss.

4. CONCLUSIONS

In this work, AdaProj a novel angular margin loss function specifically designed for semi-supervised anomaly detection with auxiliary classification tasks was presented. It was proven that this loss function learns an embedding space with class-specific subspaces of arbitrary dimension. In contrast to other angular margin losses, which try to project data to individual points in space, this relaxes the requirements of solving the classification task and allows for less compact distributions in the embedding space. In experiments conducted on the DCASE2022 and DCASE2023 ASD datasets, it was shown that using AdaProj results in better performance than other commonly used loss functions. In conclusion, the resulting embedding space has a more desirable structure than the other embedding spaces for differentiating between normal and anomalous samples. For future work, it is planned to evaluate AdaProj on other datasets and further improve the performance of the ASD system by utilizing self-supervised learning [37] or multi-task learning [9]. Investigating how the AdaProj loss performs for supervised or unsupervised tasks in comparison to other loss functions may also be of interest.

5. REFERENCES

[1] C. Aggarwal, *Outlier Analysis*, Springer, 2nd edition, 2017.

[2] K. Dohi et al., “Description and discussion on DCASE 2022 challenge task 2: Unsupervised anomalous sound detection for machine condition monitoring applying domain generalization techniques,” in *DCASE*. 2022, pp. 26–30, Tampere University.

[3] K. Dohi et al., “Description and discussion on DCASE 2023 challenge task 2: First-shot unsupervised anomalous sound detection for machine condition monitoring,” in *DCASE*. 2023, pp. 31–35, Tampere University.

[4] J. Wang, C. Lan, C. Liu, Y. Ouyang, and T. Qin, "Generalizing to unseen domains: A survey on domain generalization," in *IJCAI*. 2021, pp. 4627–4635, ijcai.org.

[5] K. Wilkinghoff and F. Kurth, "Why do angular margin losses work well for semi-supervised anomalous sound detection?," *IEEE/ACM TASLP*, vol. 32, pp. 608–622, 2024.

[6] R. Giri, S. V. Tenneti, F. Cheng, K. Helwani, U. Isik, and A. Krishnaswamy, "Self-supervised classification for detecting anomalous sounds," in *DCASE*, 2020, pp. 46–50.

[7] J. A. Lopez, H. Lu, P. Lopez-Meyer, L. Nachman, G. Stemmer, and J. Huang, "A speaker recognition approach to anomaly detection," in *DCASE*, 2020, pp. 96–99.

[8] T. Inoue et al., "Detection of anomalous sounds for machine condition monitoring using classification confidence," in *DCASE*, 2020, pp. 66–70.

[9] S. Venkatesh, G. Wichern, A. Shanmugam Subramanian, and J. Le Roux, "Improved domain generalization via disentangled multi-task learning in unsupervised anomalous sound detection," in *DCASE*. 2022, pp. 196–200, Tampere University.

[10] T. Nishida, K. Dohi, T. Endo, M. Yamamoto, and Y. Kawaguchi, "Anomalous sound detection based on machine activity detection," in *EUSIPCO*. 2022, pp. 269–273, IEEE.

[11] Y. Deng et al., "Ensemble of multiple anomalous sound detectors," in *DCASE*. 2022, Tampere University.

[12] H. Chen et al., "An effective anomalous sound detection method based on representation learning with simulated anomalies," in *ICASSP*. IEEE, 2023.

[13] D. Hendrycks, M. Mazeika, and T. G. Dietterich, "Deep anomaly detection with outlier exposure," in *ICLR*. 2019, OpenReview.net.

[14] P. Primus, V. Haunschmid, P. Praher, and G. Widmer, "Anomalous sound detection as a simple binary classification problem with careful selection of proxy outlier examples," in *DCASE*, 2020, pp. 170–174.

[15] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE SPL*, vol. 25, no. 7, pp. 926–930, 2018.

[16] L. Ruff et al., "Deep one-class classification," in *ICML*. 2018, vol. 80 of *Proceedings of Machine Learning Research*, pp. 4390–4399, PMLR.

[17] P. Perera and V. M. Patel, "Learning deep features for one-class classification," *IEEE TIP*, vol. 28, no. 11, pp. 5450–5463, 2019.

[18] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *ECCV*. Springer, 2016, pp. 499–515.

[19] J. A. Lopez, G. Stemmer, P. Lopez-Meyer, P. Singh, J. A. del Hoyo Ontiveros, and H. A. Cordourier, "Ensemble of complementary anomaly detectors under domain shifted conditions," in *DCASE*, 2021, pp. 11–15.

[20] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *CVPR*. 2019, pp. 4690–4699, IEEE.

[21] I. Kuroyanagi, T. Hayashi, Y. Adachi, T. Yoshimura, K. Takeda, and T. Toda, "An ensemble approach to anomalous sound detection based on conformer-based autoencoder and binary classifier incorporated with metric learning," in *DCASE*, 2021, pp. 110–114.

[22] K. Wilkinghoff, "Combining multiple distributions based on sub-cluster adacos for anomalous sound detection under domain shifted conditions," in *DCASE*, 2021, pp. 55–59.

[23] K. Wilkinghoff, "Design choices for learning embeddings from auxiliary tasks for domain generalization in anomalous sound detection," in *ICASSP*. 2023, IEEE.

[24] X. Zhang, R. Zhao, Y. Qiao, X. Wang, and H. Li, "AdaCos: Adaptively scaling cosine logits for effectively learning deep face representations," in *CVPR*. 2019, pp. 10823–10832, IEEE.

[25] K. Wilkinghoff, "Sub-cluster AdaCos: Learning representations for anomalous sound detection," in *IJCNN*. 2021, IEEE.

[26] Q. Yu, M. S. Kavitha, and T. Kurita, "Autoencoder framework based on orthogonal projection constraints improves anomalies detection," *Neurocomputing*, vol. 450, pp. 372–388, 2021.

[27] L. Li, Y. Zhang, and A. Huang, "Learnable subspace orthogonal projection for semi-supervised image classification," in *ACCV*. 2022, vol. 13843 of *Lecture Notes in Computer Science*, pp. 477–490, Springer.

[28] A. N. Gorban, I. Yu. Tyukin, D. V. Prokhorov, and K. I. Sofeikov, "Approximation with random bases: Pro et contra," *Information Sciences*, vol. 364-365, pp. 129–145, 2016.

[29] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015, vol. 37, pp. 448–456.

[30] N. Harada, D. Niizumi, D. Takeuchi, Y. Ohishi, M. Yasuda, and S. Saito, "ToyADMOS2: Another dataset of miniature-machine operating sounds for anomalous sound detection under domain shift conditions," in *DCASE*, 2021, pp. 1–5.

[31] K. Dohi et al., "MIMII DG: sound dataset for malfunctioning industrial machine investigation and inspection for domain generalization task," in *DCASE*. 2022, pp. 26–30, Tampere University.

[32] N. Harada, D. Niizumi, D. Takeuchi, Y. Ohishi, and M. Yasuda, "ToyADMOS2+: New toyadmos data and benchmark results of the first-shot anomalous sound event detection baseline," in *DCASE*. 2023, pp. 41–45, Tampere University.

[33] D. Katzman McClish, "Analyzing a portion of the ROC curve," *Medical decision making*, vol. 9, no. 3, pp. 190–195, 1989.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, Yoshua Bengio and Yann LeCun, Eds., 2015.

[35] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *AISTATS*. 2010, vol. 9 of *JMLR Proceedings*, pp. 249–256, JMLR.org.

[36] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," in *ICLR*. 2018, OpenReview.net.

[37] K. Wilkinghoff, "Self-supervised learning for anomalous sound detection," in *ICASSP*. 2024, pp. 276–280, IEEE.

A REFERENCE-FREE METRIC FOR LANGUAGE-QUERIED AUDIO SOURCE SEPARATION USING CONTRASTIVE LANGUAGE-AUDIO PRETRAINING

Feiyang Xiao[1], *Jian Guan*[1*], *Qiaoxi Zhu*[2], *Xubo Liu*[3], *Wenbo Wang*[4], *Shuhan Qi*[5], *Kejia Zhang*[1], *Jianyuan Sun*[6], *and Wenwu Wang*[3]

[1]College of Computer Science and Technology, Harbin Engineering University, Harbin, China
[2]University of Technology Sydney, Ultimo, Australia
[3]Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford, UK
[4]Faculty of Computing, Harbin Institute of Technology, Harbin, China
[5]School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China
[6]Department of Computer Science, University of Sheffield, Sheffield, UK

ABSTRACT

Language-queried audio source separation (LASS) aims to separate an audio source guided by a text query, with the signal-to-distortion ratio (SDR)-based metrics being commonly used to objectively measure the quality of the separated audio. However, the SDR-based metrics require a reference signal, which is often difficult to obtain in real-world scenarios. In addition, with the SDR-based metrics, the content information of the text query is not considered effectively in LASS. This paper introduces a reference-free evaluation metric using a contrastive language-audio pretraining (CLAP) module, termed CLAPScore, which measures the semantic similarity between the separated audio and the text query. Unlike SDR, the proposed CLAPScore metric evaluates the quality of the separated audio based on the content information of the text query, without needing a reference signal. Experiments show that the CLAPScore provides an effective evaluation of the semantic relevance of the separated audio to the text query, as compared to the SDR metric, offering an alternative for the performance evaluation of LASS systems. The code for evaluation is publicly available[1].

Index Terms— Language-queried audio source separation, evaluation metric, semantic similarity, CLAPScore

1. INTRODUCTION

Language-queried audio source separation (LASS) focuses on separating an audio source from a multi-source mixture based on a natural language description, i.e., a text query [1, 2]. Unlike traditional audio source separation, LASS utilizes the complex and rich semantic information of natural language to guide the separation process [1]. This integration of multi-modal data allows for more intuitive and flexible interaction with audio separation systems, making it particularly useful in various applications, i.e., audio editing [3–6], multimedia content creation [7], and designs of assistive listening devices [1, 2, 8, 9].

Following audio source separation literature [10–12], the signal-to-distortion ratio based metrics, i.e., SDR [13], SDR improvement (SDRi) [14, 15], and scale-invariant SDR (SI-SDR) [16] have been used to measure the separation performance of LASS methods in [1]. All these metrics aim to quantify the quality of the separated audio signals. They measure how close the separated audio is to the original target audio, focusing on the reduction of distortion or errors introduced during the separation process [14].

However, a major limitation of these SDR-based metrics is that they need a reference audio to compare against the separated audio. This makes these metrics applicable only in the simulated environments with known target audio, but impractical for real-world applications where the target source is unknown [17]. In such cases, alternative evaluation methods or proxy measures are required to evaluate the performance of the audio separation algorithms.

In this paper, we introduce a reference-free evaluation metric for LASS, which calculates the audio-text similarity score using the contrastive language-audio pretraining (CLAP) module [18], termed CLAPScore. Unlike the previous SDR-based metrics that require a reference audio to measure the separation performance, the proposed CLAPScore metric evaluates the semantic similarity between the separated audio and the text query without needing a reference audio. This makes CLAPScore metric particularly useful for real-world applications where a reference audio may not be available. Furthermore, similar to SDRi, the improvement in CLAPScore (CLAPScore-i) from the mixture to the separated audio can reflect the improvement from LASS methods. Moreover, the CLAPScore is also expanded to incorporate the reference audio while it is available, denoted as RefCLAPScore.

Experiments indicate that the proposed CLAPScore metric exhibits an approximately linear correlation with the SDR metric, suggesting that CLAPScore can effectively evaluate the separation performance of the LASS methods. Additionally, since the CLAPScore metric does not require reference audio and relies solely on the text query used in the LASS separation process, it can be utilized to evaluate LASS in real-world scenarios where the reference audio is unavailable. This capability facilitates the development and evaluation of the LASS methods on real-world multi-source data.

2. PREVIOUS SDR-BASED METRICS

The SDR-based metrics (i.e., SDR, SDRi, and SI-SDR) are widely used objective metrics in signal processing, particularly in language-queried audio source separation [14]. These metrics can provide a reliable and standardized method for evaluating the

*Corresponding author.

This work was partly supported by the project of the Ministry of Industry and Information Technology under Grant No.CBZ3N21-2.

[1] GitHub: https://github.com/LittleFlyingSheep/CLAPScore_for_LASS

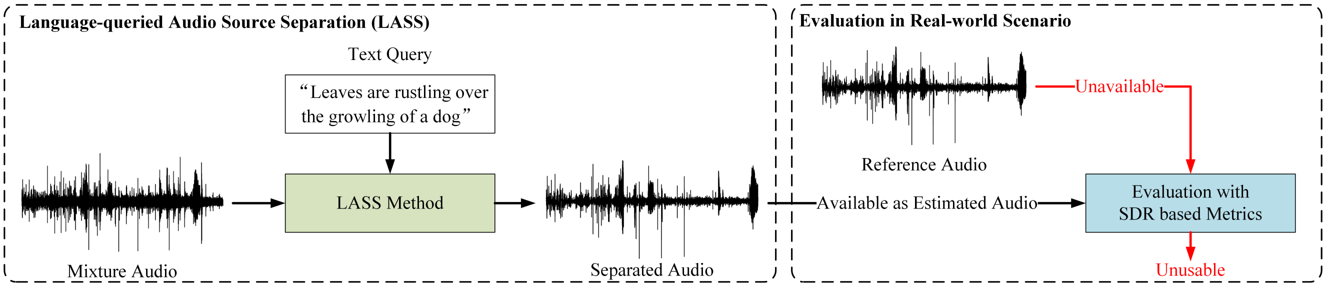

Language-queried Audio Source Separation (LASS)
Text Query
"Leaves are rustling over the growling of a dog"
LASS Method
Mixture Audio
Separated Audio
Evaluation in Real-world Scenario
Unavailable
Reference Audio
Available as Estimated Audio
Evaluation with SDR based Metrics
Unusable

Figure 1: Illustration of the limitation of the SDR-based metrics for the evaluation of the language-queried audio source separation (LASS) methods in the real-world scenario, where the reference audio required by the SDR-based metrics is unavailable. Therefore, the SDR-based metrics are unusable for the evaluation of the LASS methods in the real-world scenario.

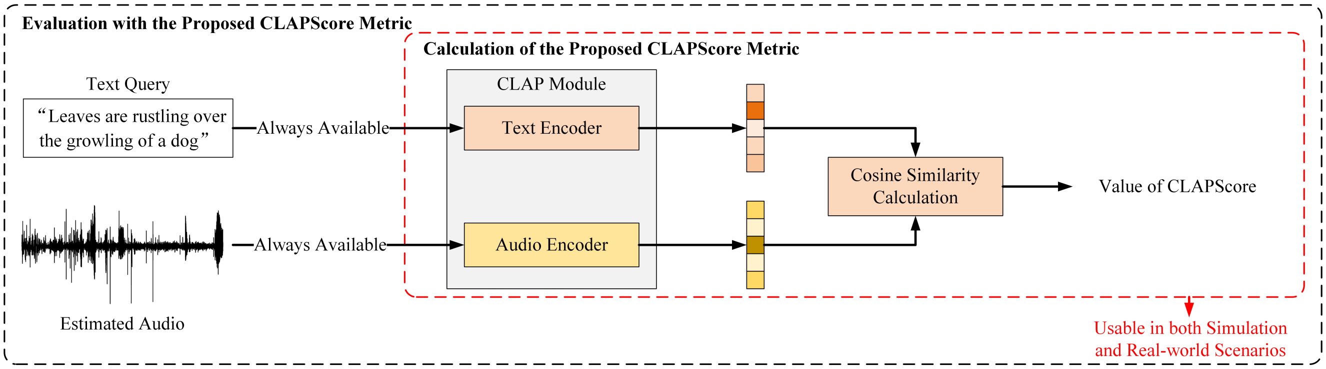

Evaluation with the Proposed CLAPScore Metric
Calculation of the Proposed CLAPScore Metric
Text Query
"Leaves are rustling over the growling of a dog"
Always Available
CLAP Module
Text Encoder
Audio Encoder
Always Available
Estimated Audio
Cosine Similarity Calculation
Value of CLAPScore
Usable in both Simulation and Real-world Scenarios

Figure 2: Illustration of the evaluation process with the proposed CLAPScore metric for language-queried audio source separation. Notably, the proposed CLAPScore metric does not need a reference audio for the evaluation. The inputs of the proposed CLAPScore metric, i.e., the estimated audio and the text query, are available in both simulation and real-world scenarios. Therefore, the CLAPScore metric can be applicable for both such scenarios.

quality of the separated audio from LASS methods in the simulation scenario but are limited in the real world [17].

2.1. Definition of SDR-Based Metrics

In widely used SDR-based metrics, SDR measures the ratio of the power of the desired signal to the power of the distortion introduced by the separation process [13]. SDRi is an improvement metric that measures the difference in SDR before and after applying an audio source separation algorithm [14, 15]. SI-SDR normalizes the audio signals to make the evaluation independent of their amplitude, which is more robust for varying scales [16, 19, 20]. The definition of SDR, SDRi and SI-SDR can be presented as follows:

$$\text{SDR} = 10\log_{10}\left(\frac{\|\mathbf{s}\|^2}{\|\mathbf{s}-\hat{\mathbf{s}}\|^2}\right), \tag{1}$$

$$\text{SDRi} = \text{SDR}_{\text{after}} - \text{SDR}_{\text{before}}, \tag{2}$$

$$\text{SI-SDR} = 10\log_{10}\left(\frac{\|\alpha\mathbf{s}\|^2}{\|\alpha\mathbf{s}-\hat{\mathbf{s}}\|^2}\right), \tag{3}$$

where $\mathbf{s}$ denotes the reference audio, i.e., the ground-truth audio source, $\hat{\mathbf{s}}$ denotes the estimated audio. $\text{SDR}_{\text{before}}$ denotes the SDR between the mixture and the reference audio, and $\text{SDR}_{\text{after}}$ denotes the SDR between the separated audio from a LASS method and the reference audio. The improvement from $\text{SDR}_{\text{before}}$ to $\text{SDR}_{\text{after}}$ is the value of SDRi. For SI-SDR, $\alpha = \frac{\hat{\mathbf{s}}^\top \mathbf{s}}{\|\mathbf{s}\|^2}$ is the optimal scaling factor that aligns the estimated audio with the reference audio, where $\top$ denotes the transpose operation. For all of these SDR-based metrics, a higher value indicates better separation performance.

2.2. Limitation of SDR-Based Metrics

According to the above definition of SDR-based metrics, it can be found that, these metrics all depend on the reference audio signal $\mathbf{s}$ to measure the separation performance of the LASS methods. However, this requirement can be only met in a simulation scenario, where the reference audio and the noise are known to simulate the mixture audio. Due to the lack of the reference audio, these SDR-based metrics cannot be usable to measure the LASS performance in the real-world scenario, as illustrated in Figure 1.

Moreover, these SDR-based metrics are power-based metrics to measure the effectiveness of LASS methods. They primarily focus on the signal quality and distortion level of the separated audio, without considering whether the semantic content of the separated audio matches the text query. Therefore, these SDR-based metrics cannot measure the semantic similarity between the separated audio and the text query. To measure the matching of the semantic content between the separated audio and the text query, other more effective semantic similarity metrics are required.

3. PROPOSED CLAPSCORE METRIC

To measure how well the separated audio matches the text queries, we introduce the CLAPScore metric. This metric quantifies how closely the content of the separated audio aligns with the text query. A higher CLAPScore means that the separated audio's content is more similar to the text query, indicating better performance in separating audio based on the text query. The evaluation process with the proposed CLAPScore metric is illustrated in Figure 2.

3.1. Definition of Proposed CLAPScore Metric

The proposed CLAPScore metric is a measure of the similarity between the separated audio from the LASS methods and the text query used in the LASS process. It can measure the semantic similarity between the separated audio and the text query.

The calculation of the proposed CLAPScore metric is based on the contrastive language-audio pretraining (CLAP) module [18]. The CLAP module is pretrained on a large-scale dataset and learns the audio-text alignment in the latent space [18]. Due to this advantage, the CLAP module is widely used to measure the audio-text alignment in the evaluation of text-to-audio generation methods [21, 22]. Inspired by these studies, we introduce the CLAP module to calculate the audio-text similarity between the estimated audio and the text query to measure the separation performance of the LASS methods.

Specifically, the audio embedding of the estimated audio $\hat{\mathbf{s}}$ (i.e., the separated audio signal) and the text embedding of the text query are obtained with the CLAP module[2], as follows,

$$\hat{\mathbf{a}} = E_A(\hat{\mathbf{s}}), \tag{4}$$

$$\mathbf{t} = E_T(\mathbf{c}), \tag{5}$$

where $\mathbf{c}$ denotes the text query, $E_A(\cdot)$ and $E_T(\cdot)$ denotes the audio encoder and text encoder in CLAP module, respectively. The audio embedding $\hat{\mathbf{a}}$ of the estimated audio is extracted by the audio encoder in the CLAP module, and the text embedding $\mathbf{t}$ of the text query is extracted by the text encoder in the CLAP module.

Then, the cosine similarity between the audio embedding and the text embedding is calculated as the value of the proposed CLAPScore metric to measure the semantic similarity between the estimated audio and the text query. Thus, the calculation of the audio-text similarity score can be represented as

$$\text{CLAPScore} = \frac{\hat{\mathbf{a}}^\top \mathbf{t}}{||\hat{\mathbf{a}}|| \, ||\mathbf{t}||}. \tag{6}$$

A higher CLAPScore means a better match between the audio embedding of the estimated audio and the text query used in LASS process. Therefore, a higher CLAPScore indicates better separation performance of the LASS methods.

3.2. Advantages of the Proposed CLAPScore Metric

Different from the SDR-based metrics, the proposed CLAPScore metric can evaluate the degree of matching between the separated audio and the text query in their latent spaces. It provide a way to measure the semantic similarity between the separated audio and the text query for the LASS task.

In addition, according to the definition of the proposed CLAPScore metric, it can be found that, the evaluation based on the proposed CLAPScore metric depends on the separated audio and the text query, without the need for a reference audio as required in the SDR-based metrics. The separated audio and the text query can be easily obtained in both the simulation and the real-world scenarios, thus this metric is applicable for both scenarios, offering advantages over the SDR-based metrics which only work when the reference audio is available.

3.3. Expanded CLAPScore Improvement Metric

In addition, similar to the SDRi metric, we design the improvement of the CLAPScore metric to measure the difference in the proposed CLAPScore metric before and after applying an LASS method, termed CLAPScore improvement (CLAPScore-i). The CLAPScore-i metric can be calculated as follows,

$$\text{CLAPScore-i} = \text{CLAPScore}_{\text{after}} - \text{CLAPScore}_{\text{before}}, \tag{7}$$

where $\text{CLAPScore}_{\text{before}}$ denotes the CLAPScore between the original mixture audio and the text query, and $\text{CLAPScore}_{\text{after}}$ denotes the CLAPScore between the separated audio and the text query.

3.4. Expanded RefCLAPScore Metric

We present an expanded CLAPScore while the reference audio is available, termed RefCLAPScore. The calculation of the RefCLAPScore can be represented as

$$\text{RefCLAPScore} = H(\text{CLAPScore}_{\text{after}}, \text{CLAPScore}_{\text{ref}}), \tag{8}$$

where $H(\cdot,\cdot)$ denotes the harmonic mean function, and $\text{CLAPScore}_{\text{ref}}$ denotes the CLAPScore of the reference audio. The RefCLAPScore metric can further introduce the semantic information of the reference audio (i.e., source audio) to obtain a fine-grained measure for the separation performance.

4. EXPERIMENTS

4.1. Dataset

To verify the effectiveness of the proposed CLAPScore metric, we conducted experiments on the DCASE 2024 Challenge Task 9 validation set[3]. This dataset includes 1000 audio signals from the FreeSound dataset [23], each with 3 corresponding text queries. By randomly combining pairs of audio signals, the validation set provides 3000 mixture audio samples for evaluation. Additionally, we split this dataset to perform an ablation study of the proposed CLAPScore metric.

4.2. Effectiveness of Proposed CLAPScore Metric

To demonstrate the effectiveness of the proposed CLAPScore metric, we evaluate the separation performance of standard LASS methods on 3000 officially provided mixture audio signals using both SDR-based metrics (SDR, SDRi, SI-SDR) and CLAPScore based metrics (CLAPScore, CLAPScore-i, RefCLAPScore). The evaluated LASS methods include the official baseline of the DCASE 2024 Challenge Task 9 (baseline) [2], our previously submitted system [24] trained with GPT-augmented text queries (baseline-Augmented) [25,26], and the state-of-the-art method, AudioSep [2]. Evaluation results measured by these metrics are shown in Table 1.

Based on the SDR metric performance, it is clear that the separation effectiveness of the three evaluated LASS methods ranks from highest to lowest as follows: AudioSep, baseline-Augmented, and baseline. Similarly, in the evaluation using the CLAPScore metric, the methods rank from best to worst in the same order: AudioSep, baseline-Augmented, and baseline. This demonstrates that the CLAPScore metric can effectively assess the separation performance of LASS methods. Furthermore, its ability to evaluate without requiring a reference audio makes it particularly suitable for scenarios where reference audio is unavailable.

[2] https://huggingface.co/spaces/Audio-AGI/AudioSep/blob/main/checkpoint/music_speech_audioset_epoch_15_esc_89.98.pt

[3] https://zenodo.org/records/10886481

Table 1: Evaluation of different LASS methods with the SDR-based metrics (i.e., SDR, SDRi, SI-SDR) and the proposed CLAPScore based metrics (i.e., CLAPScore, CLAPScore-i, RefCLAPScore).

Method	SDR	SDRi	SI-SDR	CLAPScore	CLAPScore-i	RefCLAPScore
Baseline [2]	5.708	5.673	3.862	0.239	0.029	0.253
Baseline-Augmented [24]	5.937	5.902	4.191	0.242	0.031	0.254
AudioSep [2]	**8.192**	**8.157**	**6.680**	**0.261**	**0.050**	**0.267**

Table 2: Pearson correlation coefficient (PCC) between SDR-based and CLAPScore-based metrics with statistically significant correlation p-value < 0.05.

	PCC with SDR	PCC with SI-SDR		PCC with SDRi
CLAPScore	0.270	0.289	CLAPScore-i	0.288
RefCLAPScore	0.226	0.254		

4.3. Correlation between SDR-Based Metrics and CLAPScore

According to the results in Table 1, an interesting phenomenon can be observed that the performance measured by CLAPScore based metrics (i.e., CLAPScore, CLAPScore-i, and RefCLAPScore) shows similar trend to that measured by SDR-based metrics. Specifically, when the performance on CLAPScore based metrics is high, the performance on SDR-based metrics is also high. To explore their correlation, we calculate the Pearson correlation coefficient (PCC) as Table 2.

It can be found that, both CLAPScore and RefCLAPScore shows a moderate positive correlation with both SDR and SI-SDR. Additionally, CLAPScore-i has a similar moderate correlation with SDRi. These indicate that the CLAPScore based metrics has statistically significant positive correlations with SDR-based metrics.

To further explore the correlation between these metrics, we simulate the mixture audio under different SDR levels ranging from −20dB to 20dB in 5dB increments, based on the provided 3000 source-noise pairs in the validation set of DCASE 2024 Challenge Task 9. Then, we evaluate the quality of these simulated mixture audio and the quality of the separated audio from the LASS method (i.e., AudioSep [2]) using the proposed CLAPScore based metrics. The results are illustrated in Figure 3.

The proposed CLAPScore for mixture audio shows an approximately linear correlation with the SDR metric, as shown by the blue line in Figure 3. This indicates that CLAPScore effectively evaluates audio signal quality using text queries. Additionally, Figure 3 demonstrates that the CLAPScore for separated audio (red line) and CLAPScore-i (green line) indicate a better match with text queries for separated audio, validating CLAPScore's effectiveness in measuring separated audio quality. Notably, CLAPScore-i for AudioSep is higher at lower SDR levels than at higher SDR levels, likely because simulated mixtures at higher SDR levels are already close to the source audio, resulting in only subtle improvements with the LASS method.

4.4. Evaluation with Different Mixing Strategies

We conduct an ablation study to evaluate the CLAPScore value of the mixture audio signals with different mixing strategies, where 990 audio signals are selected from the validation set of DCASE 2024 Challenge Task 9 as source audio and three different mixing strategies are attempted for each source audio: (1) source audio, (2) mixed with white noise, and (3) mixed with an audio signal of different content. This results in a total of 2970 mixtures for evaluation, with each mixing strategy producing 990 estimated audio signals. The lines representing the CLAPScore metric at different SDR levels (−20dB, −15dB, −10dB, −5dB, 0dB, 5dB, 10dB, 15dB, and 20dB) for these mixtures are shown in Figure 4.

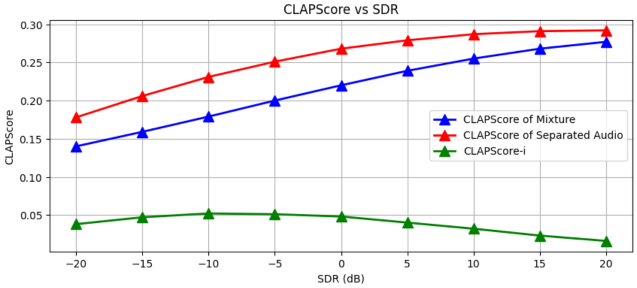


Figure 3: Illustration to show the correlation between the SDR metric and the proposed CLAPScore metric. Here, the separated audio comes from the LASS method, i.e., AudioSep [2].

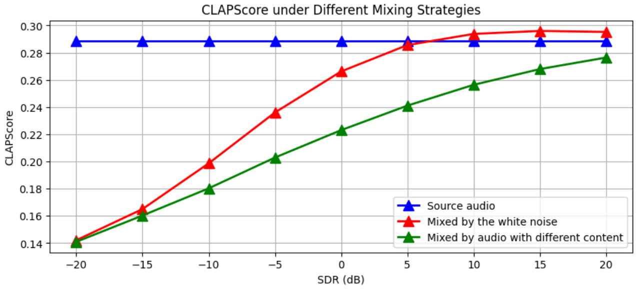


Figure 4: Illustration of the proposed CLAPScore metric for the mixtures from different mixing strategies.

It can be found that, the value of the proposed CLAPScore for the source audio is significantly better than the one mixed by audio with different content, under any SDR levels. This verifies that the proposed CLAPScore metric can capture the difference on the semantic content between the estimated audio and the text query. Therefore, the proposed CLAPScore metric prefers to assign an estimated audio that has different content from the text query with a lower measure, even if the SDR performance of the estimated audio is good (i.e., 20dB).

Furthermore, it is interesting that the estimated audio mixed with the white noise has higher CLAPScore value than the original source audio under high SDR levels (i.e., 10dB, 15dB, 20dB). The reason may be that, in these SDR levels, the white noise can be considered as the background noise, estimated audio mixed by such background noise may enhance the realism of the resulting mixes, as analyzed in [9]. Then, the enhanced realism of the estimated audio leads to better CLAPScore performance than the source audio.

5. CONCLUSION

In this work, we proposed a reference-free metric for language-queried audio source separation using contrastive language-audio pretraining, termed CLAPScore, which can further measure the semantic similarity between the estimated audio and the text query, without the requirement of a reference audio. Experiments show that the proposed CLAPScore can achieve a more fine-grained evaluation for language-queried audio source separation.

6. REFERENCES

[1] X. Liu, H. Liu, Q. Kong, X. Mei, J. Zhao, Q. Huang, M. D. Plumbley, and W. Wang, "Separate what you describe: Language-queried audio source separation," in *Proc. INTERSPEECH*, 2022.

[2] X. Liu, Q. Kong, Y. Zhao, H. Liu, Y. Yuan, Y. Liu, R. Xia, Y. Wang, M. D. Plumbley, and W. Wang, "Separate anything you describe," *arXiv preprint arXiv:2308.05037*, 2023.

[3] Y. Wang, H. Chen, D. Yang, J. Yu, C. Weng, Z. Wu, and H. Meng, "Consistent and relevant: Rethink the query embedding in general sound separation," in *Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2024, pp. 961–965.

[4] Y. Liu, X. Liu, Y. Zhao, Y. Wang, R. Xia, P. Tain, and Y. Wang, "Audio prompt tuning for universal sound separation," in *Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2024, pp. 1446–1450.

[5] R. Tan, A. Ray, A. Burns, B. A. Plummer, J. Salamon, O. Nieto, B. Russell, and K. Saenko, "Language-guided audio-visual source separation via trimodal consistency," in *Proc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023, pp. 10 575–10 584.

[6] H.-W. Dong, N. Takahashi, Y. Mitsufuji, J. McAuley, and T. Berg-Kirkpatrick, "CLIPSep: Learning text-queried sound separation with noisy unlabeled videos," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2023.

[7] H. Liu, Z. Chen, Y. Yuan, X. Mei, X. Liu, D. Mandic, W. Wang, and M. D. Plumbley, "AudioLDM: Text-to-audio generation with latent diffusion models," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2023.

[8] Q. Kong, Y. Cao, H. Liu, K. Choi, and Y. Wang, "Decoupling magnitude and phase estimation with deep ResUNet for music source separation," in *Proc. Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, 2021, pp. 342–349.

[9] J. Pons, X. Liu, S. Pascual, and J. Serrà, "GASS: Generalizing audio source separation with large-scale data," in *Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2024, pp. 546–550.

[10] Y. Luo and N. Mesgarani, "TasNet: Time-domain audio separation network for real-time, single-channel speech separation," in *Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2018, pp. 696–700.

[11] F. Xiao, J. Guan, Q. Kong, and W. Wang, "Time-domain speech enhancement with generative adversarial learning," *arXiv preprint arXiv:2103.16149*, 2021.

[12] Y. Luo and N. Mesgarani, "Conv-TasNet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 27, no. 8, pp. 1256–1266, 2019.

[13] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE Trans. Audio Speech Lang. Process.*, vol. 14, no. 4, pp. 1462–1469, 2006.

[14] R. Scheibler, "SDR–medium rare with fast computations," in *Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2022, pp. 701–705.

[15] K. Chen, J. Su, and Z. Jin, "MDX-GAN: Enhancing perceptual quality in multi-class source separation via adversarial training," in *Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2024, pp. 741–745.

[16] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "SDR–half-baked or well done?" in *Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2019, pp. 626–630.

[17] S. Montrésor, P. Picart, and M. Karray, "Reference-free metric for quantitative noise appraisal in holographic phase measurements," *J. Opt. Soc. Am. A*, vol. 35, no. 1, pp. A53–A60, 2018.

[18] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, "Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation," in *Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2023, pp. 1–5.

[19] E. Tzinis, S. Venkataramani, Z. Wang, C. Subakan, and P. Smaragdis, "Two-step sound source separation: Training on learned latent targets," in *Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2020, pp. 31–35.

[20] T. von Neumann, K. Kinoshita, C. Boeddeker, M. Delcroix, and R. Haeb-Umbach, "SA-SDR: A novel loss function for separation of meeting style data," in *Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2022, pp. 6022–6026.

[21] R. Huang, J. Huang, D. Yang, Y. Ren, L. Liu, M. Li, Z. Ye, J. Liu, X. Yin, and Z. Zhao, "Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models," in *Proc. Int. Conf. Mach. Learn. (ICML)*. PMLR, 2023, pp. 13 916–13 932.

[22] H. Liu, Y. Yuan, X. Liu, X. Mei, Q. Kong, Q. Tian, Y. Wang, W. Wang, Y. Wang, and M. D. Plumbley, "AudioLDM 2: Learning holistic audio generation with self-supervised pretraining," *IEEE/ACM Trans. Audio Speech Lang. Process.*, 2024.

[23] E. Fonseca, J. Pons Puig, X. Favory, F. Font Corbera, D. Bogdanov, A. Ferraro, S. Oramas, A. Porter, and X. Serra, "Freesound datasets: A platform for the creation of open audio datasets," in *Proc. Int. Soc. Music Inf. Retr. (ISMIR)*, 2017, pp. 486–493.

[24] F. Xiao, W. Wang, D. Xu, S. Qi, Q. Zhu, and J. Guan, "Language-queried audio source separation with GPT-based text augmentation and ideal ratio masking," DCASE2024 Challenge, Tech. Rep., June 2024.

[25] P. Primus, K. Koutini, and G. Widmer, "CP-JKU's submission to task 6b of the DCASE2023 challenge: Audio retrieval with PaSST and GPT-augmented captions," DCASE2023 Challenge, Tech. Rep., June 2023.

[26] P. Primus, K. Koutini, and G. Widmer, "Advancing natural-language based audio retrieval with PaSST and large audio-caption data sets," in *Proc. Detect. Classif. Acoust. Scenes Events (DCASE) Workshop*, Tampere, Finland, September 2023, pp. 151–155.

WILDDESED: AN LLM-POWERED DATASET FOR WILD DOMESTIC ENVIRONMENT SOUND EVENT DETECTION SYSTEM

Yang Xiao and Rohan Kumar Das

Fortemedia Singapore, Singapore
{xiaoyang, rohankd}@fortemedia.com

ABSTRACT

This work aims to advance sound event detection (SED) research by presenting a new large language model (LLM)-powered dataset namely wild domestic environment sound event detection (WildDESED). It is crafted as an extension to the original DESED dataset to reflect diverse acoustic variability and complex noises in home settings. We leveraged LLMs to generate eight different domestic scenarios based on target sound categories of the DESED dataset. Then we enriched the scenarios with a carefully tailored mixture of noises selected from AudioSet and ensured no overlap with target sound. We consider widely popular convolutional neural recurrent network to study WildDESED dataset, which depicts its challenging nature. We then apply curriculum learning by gradually increasing noise complexity to enhance the model's generalization capabilities across various noise levels. Our results with this approach show improvements within the noisy environment, validating the effectiveness on the WildDESED dataset promoting noise-robust SED advancements.

Index Terms— sound event detection, DESED, noisy scenario, noise robust SED, curriculum learning

1. INTRODUCTION

Sounds play a vital role in our lives, helping us understand our surroundings and notice changes. Sound event detection (SED) [1, 2] is essential for interpreting and responding to our environment, with applications ranging from urban noise management to smart-home technologies [3] and security systems [4]. SED has made great strides [5–7], thanks to diverse datasets [8] tailored for specific scenarios. Google AudioSet [9] provides a wide array of sounds, and MAVD [10] focuses on traffic noise. Among various SED datasets, DESED [11, 12] is well known for its focus on domestic environments, which makes it the most utilized dataset for home sound event research. However, DESED faces challenges in comprehensively representing the unpredictable and complex nature of household sounds. Hence, there exists scope for covering a wide range of domestic scenarios with common background noises that can occur in a household.

The quest for noise robustness in SED has led to the development of new methodologies and datasets [13, 14] aimed towards improving performance under challenging conditions such as noisy urban environments. Innovations by researchers like Neri et al. [15], Serizel et al. [16], and Wan et al. [17] have pushed the boundaries of SED systems by integrating deep learning and audio enhancement techniques. These studies, however, predominantly address controlled or semi-controlled environments, leaving a gap for SED systems to effectively detect sound events in the less predictable, 'wild' conditions in domestic environments.

Addressing this gap, our research contributes to the field by introducing a new dataset namely, *wild domestic environment sound event detection* (WildDESED). We proposed carefully selecting noise types from the AudioSet that accurately represent real home environments but are distinct from DESED's target sounds. This artificial selection could be challenging because of the bias and unnatural correlations. Large language models (LLMs) [18] such as GPT-4, ChatGPT, and Llama have demonstrated remarkable potential to perform various tasks [19–21] in recent years. In this regard, we utilized the strong capabilities of LLMs to analyze and summarize acoustic data for selecting specific noises. This helped us to design eight different scenarios that blend the noises with target sounds, simulating authentic domestic environments. The noises are divided into four categories based on their sources and acoustic properties, allowing for a diverse and realistic combination with target sounds. This novel approach has culminated in the creation of WildDESED dataset, specifically designed to enhance SED research in dynamic and natural home environments.

Building on this foundation, our research not only introduces the WildDESED dataset, but also explores the application of curriculum learning in the context of SED to tackle the challenges posed by domestic noisy environments. Curriculum learning [22–24] is a training approach that improves models for noisy speech [25–27] and audio by starting with simpler, less noisy data and gradually increasing the noise level. This method is similar to the way how the humans learn and helps models adjust from clean to noisy sounds more effectively. In this work, we applied curriculum learning to the baseline convolutional recurrent neural network (CRNN) [28–30] model using the WildDESED dataset for our studies. The novelty of this work lies in the proposal of a new *in-the-wild* dataset for advancing SED research and exploring curriculum learning as an approach to develop noise-robust SED systems. The WildDESED dataset has been made publicly available[1].

2. RELATED WORK

The WildDESED dataset is an extension to the original DESED dataset, which is a foundational resource featuring 10 target sound classes pivotal for understanding the sounds in home environments. The DESED dataset consists of the following subsets: The weak set, with 1,578 real recordings labeled with weak annotations, captures the presence of sound classes without temporal specifics. The unlabeled training set includes 14,412 real, unlabeled recordings. The test set comprises of 1,168 real recordings with strong annotations to assess model performance. These three subsets are real-world recordings from AudioSet. The training synth set contains 10,000 synthetic recordings with strong annotations [31], detailing

[1] https://github.com/swagshaw/WildDESED

Table 1: A summary of different background noises used in WildDESED dataset.

Noise	Occurrences	Duration (Second)
Bird chirping outside	9,847	7,523
Car passing by outside	311	862
Chair moving	343	359
Clock ticking	2,5777	2,662
Coffee machine	6	30
Door closing	335	196
Fan noise	117	958
Footsteps	6,243	2,101
Light rain	159	1,379
Refrigerator humming	58	456
TV playing in the background	805	7,191
Wind blowing	5,467	48,648
Total	49,468	72,365

exact temporal boundaries. The synth validation set has 2,500 synthetic recordings with strong annotations for model validation during development. These two synthetic subsets are generated with the Scaper. Their background files are extracted from SINS [32], TUT [33], MUSAN [34], or YouTube and have been selected because they contain a very low amount of our sound event classes. We propose to simulate more diverse and complex noisy scenarios that are not covered by the original DESED dataset and also introduce a controlled variability for testing.

3. WILDDESED

We extend DESED to the WildDESED for in-the-wild scenarios for domestic environments by considering three primary set of questions to address as follows:

- What type of background noises do we use?
- What are the domestic scenarios we choose?
- How do we mix the background noises to the scenarios?

GPT-4 is an advanced language model that builds on the GPT-3 architecture but uses a larger amount of training data. It includes the latest techniques to enhance understanding of natural language. In the following subsections, we will detail how we leverage GPT-4 to address each of these questions, outlining the methodology behind the creation of the WildDESED dataset. This new dataset aims to bridge the gap between the controlled environment of existing datasets and the dynamic, often unpredictable nature of real-world domestic soundscapes, thus expanding the potential for noise-robust SED research in truly 'wild' home scenarios.

3.1. What type of background noises do we use?

To construct the WildDESED dataset, we initiated our process with the foundational DESED dataset, which identifies 10 distinct sound events in 10-second audio clips. The events in DESED include diverse household sounds like alarms, appliances, pets, and running water. We input the total 356 classes from the strongly annotated subset of AudioSet to the GPT-4 together with the 10 DESED classes. Then we guide GPT-4 by the following prompt:

"Select noise classes from the 356 strongly annotated AudioSet classes, alongside the 10 DESED classes ensuring clear delineation and no overlap with DESED's sound events. Further, apply thorough filtering to exclude any AudioSet classes similar to DESED target classes, preserving the distinctiveness of the dataset."

Considering the output of GPT-4, we enhanced DESED with selected events from the strongly annotated subset of AudioSet, ensuring clear delineation and no overlap with DESED's sound events. A thorough filtering process was applied to exclude any AudioSet classes that are very similar to target classes of DESED dataset, preserving the distinctiveness of our dataset. Table 1 displays the outcome of our selection process, listing the types and quantities of noise clips integrated into WildDESED. We included a spectrum of sounds both indoor, like the clock ticking, and outdoor, such as birds chirping that capture the essence of a domestic environment. The 'clock ticking' class, for instance, has the largest event count, while 'wind blowing' spans the greatest duration, together reflecting the continuous and transient nature of home sounds.

This dataset construction ensures WildDESED encompasses a rich and authentic array of domestic noises, ready to challenge and advance SED systems in recognizing the events under complex acoustic home environments.

3.2. What are the domestic scenarios we choose?

For the WildDESED dataset, we still have to map the selected 12 noise classes with our 10 target classes. We input them to GPT-4 and use the following prompt:

"Create eight different domestic scenarios so that they should map 12 selected noise classes to the 10 target classes from the DESED dataset, crafting authentic household soundscapes. Ensure the scenarios reflect typical sounds one would encounter in a household environment."

Considering the output of LLM, we crafted eight different domestic scenarios, each mapping to target classes from the DESED dataset to create authentic soundscapes one would encounter in a household. These scenarios are constructed to reflect the typical activities and the accompanying sounds in a domestic environment.

- **Morning Routine:** Associated with *'Blender'* target sounds, this scenario captures the essence of the morning with 'Light rain', 'Refrigerator humming', 'Clock ticking', and 'TV playing in the background'.
- **Home Office:** Linked to *'Speech'* as the target class, it includes background sounds of 'Car passing by', 'Fan noise', and 'Footsteps', emulating a work-from-home setting.
- **Household Chores:** Representing *'Vacuum cleaner'* noises as the target, this scenario combines 'Door closing', 'Chair moving', and 'Footsteps' as background to depict cleaning activities.
- **Late-night:** Tied to the *'Electric shaver toothbrush'* target sound, offering the 'Clock ticking' and 'Light rain' as a backdrop for night-time routines.
- **Cooking:** Merging the target sounds of *'Frying'* and *'Dishes'* with 'Coffee machine' buzzes and 'Refrigerator humming', this scenario is bustling with culinary activity.
- **Pet Care:** Incorporating target sounds of *'Cat'* and *'Dog'*, this setting is further brought to life with 'Bird chirping outside' and 'TV playing in the background'.
- **Bathroom Routine:** Linked to *'Running water'* as the target sound, with added 'Fan noise' and 'Wind blowing', simulating personal care sounds.
- **Emergency:** Associated with the *'Alarm bell ringing'* target sound, it layers urgent sounds like 'Refrigerator humming' and 'Fan noise' with 'Clock ticking' and 'Car passing by'.

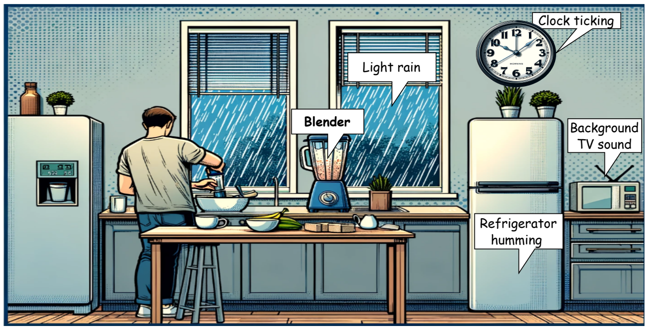


Figure 1: Illustration[3]of Morning Routine Scenario out of the total eight scenarios in WildDESED dataset. In the scenario, key target sound events are written in bold fonts, along with added different background noises to simulate real-life settings.

Each scenario's sound design is a thoughtful blend of target and noise classes, chosen to challenge the detection capabilities of SED systems within the rich and varied auditory context of a home environment. To illustrate our scenarios, we present a Figure 1 that showcases two typical scenarios out of the eight: the 'Pet Care Scenario' and the 'Morning Routine Scenario'. This figure highlights key target sound events within each scenario, incorporating strategically placed background noises to simulate the real-life acoustic challenges found in domestic settings.

3.3. How do we mix the background noises to the scenarios?

In the WildDESED dataset, the integration of background noises into the selected domestic scenarios is meticulously structured around a quadrant based on the acoustic characteristics of the noises. The quadrant categorizes noises into four groups: Ambient Environmental Sounds, Human-related and Intermittent Sounds, Mechanical and Electronic Sounds, and Nature and Outdoor Sounds, as illustrated in Figure 2.

- For **Ambient Environmental Sounds**, such as 'Light rain' and 'Wind blowing', we repeated these sounds to cover the entire duration of the audio clip from the original DESED dataset. These sounds are mixed at a low intensity to ensure they provide a consistent background atmosphere without overpowering the primary sound events. The rationale behind this is to create an unobtrusive ambient layer that emulates the continuous presence of these sounds in a typical home environment.
- Sounds like 'Footsteps', 'Door closing', and 'Chair moving' fall into the **Human-Related and Intermittent Sounds** category. These are inserted at random intervals to simulate the sporadic nature of human movement and activities within a home. The volume and frequency of these sounds are varied ±10% range to reflect the realistic and unpredictable nature of their occurrence in daily life.
- **Mechanical sounds**, including 'Clock ticking' and 'Coffee machine', are inserted at specific points to coincide with the actions they represent, such as a coffee machine being used during morning routines. The volume is set to be noticeable but not overwhelming, ensuring the sound is recognized as a part of the scenario without becoming a large distraction.
- Lastly, **Nature and Outdoor Sounds** like 'Car passing by outside' and 'Bird chirping outside' are incorporated randomly to enhance the realism of external environmental influences. The volume may fluctuate to mimic the variable volume of these sounds in real settings, contributing to the unpredictability and diversity of the overall soundscape.

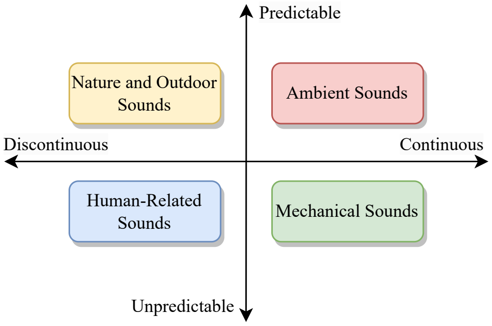


Figure 2: Quadrant showing four groups of noise types based on their acoustic characteristics considered in the WildDESED.

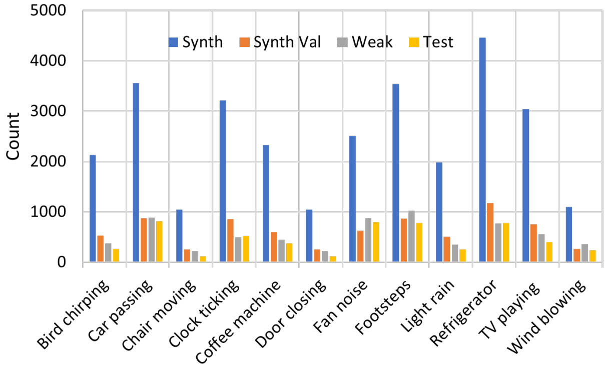


Figure 3: Statistics of noises in the WildDESED subsets.

Each noise type and its corresponding mixing approach are tailored to maintain the authenticity of the domestic scenarios. This methodical and scenario-specific approach to mix noises ensures that the WildDESED dataset not only presents a challenge for SED systems but also closely reflects the complex acoustic environments of actual domestic settings.

In finalizing the composition of the WildDESED dataset, special consideration was given to the representation of the 'speech' sound class due to its prevalence and significance in domestic environments. For the 'Home Office' scenario in synth set and synth val set, we exclusively selected clips that featured the 'speech' class in isolation, omitting any clips where 'speech' occurred alongside other sound events.

Figure 3 displays class-wise statistics for different background noises in each subset of the WildDESED dataset, indicating the prevalence of each noise type, within synth, synth val, weak, and test subsets. Figure 4 shows scenario-wise statistics for the scenarios in the WildDESED dataset, quantifying how frequently each scenario appears in each subset. Through this detailed dataset structure, WildDESED dataset positions itself as a crucial resource for developing and evaluating SED systems, equipping researchers with the means to advance the field of SED in diverse naturalistic home environments.

[3] Figures generated using DALL-E-2 (https://openai.com/dall-e-2)

Table 2: Performance in PSDS1 (P1), PSDS2 (P2) and PSDS1 + PSDS2 (P1 + P2) of the proposed curriculum learning (CL) approach on the DESED devtest set and our proposed WildDESED (W) dataset with SNR in dB.

Model	**Performance on DESED**			**Performance on WildDESED**											
				10dB			5dB			0dB			-5dB		
	P1	P2	P1 + P2	P1	P2	P1 + P2	P1	P2	P1 + P2	P1	P2	P1 + P2	P1	P2	P1 + P2
CRNN	0.344	0.543	**0.887**	0.222	0.409	0.631	0.148	0.302	0.450	0.064	0.174	0.238	0.017	0.078	0.095
CRNN (W)	0.200	0.329	0.529	0.175	0.337	0.512	0.135	0.303	0.438	0.087	0.242	0.329	0.048	0.174	0.222
CRNN (W+ CL)	0.265	0.461	0.726	0.212	0.443	**0.655**	0.175	0.390	**0.565**	0.114	0.317	**0.431**	0.049	0.211	**0.260**

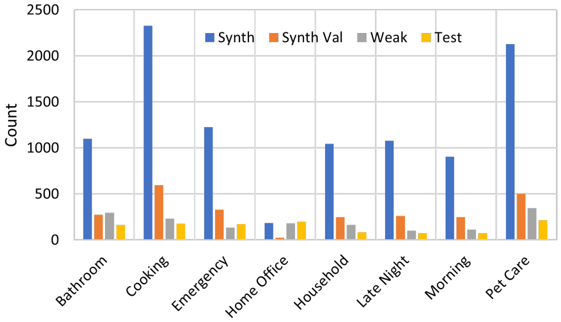


Figure 4: Statistics of the scenarios in the WildDESED subsets.

4. CURRICULUM LEARNING FOR NOISE-ROBUST SED

We use a curriculum learning [22, 26] method to develop noise-robust SED systems. This approach introduces complexity in stages, starting with simple tasks and gradually integrating noise at various signal-to-noise ratios (SNR), aligning with our goal to augment the model's resilience to noise.

We have five stages in our methodology, each with an increasing level of noise difficulty. Initially, the model learns from clean audio samples. This foundational step is crucial for establishing an understanding of the sound events without the confounding presence of noise. We then incrementally introduce noise, decreasing the SNR by 5dB in subsequent stages. Let N be the total number of training samples. Given k noise levels $L = [L_1, L_2, \ldots, L_k]$, the dataset D is composed as follows:

$$D = \bigcup_{i=1}^{k} \{D_i\}, D_i = \frac{N}{k} \text{ samples at noise level } L_i \qquad (1)$$

The k in our experiment here is 5 including the clean DESED, and noise levels 10dB, 5dB, 0dB, and -5dB are considered. The model's progress is meticulously monitored, and a validation metric c is used to evaluate learning at each epoch. In our approach, the c is the intersection f1-score. If c fails to improve for ten consecutive epochs [35], the best-performing model state is reloaded, and the training progresses to the next noise level.

5. EXPERIMENTAL SETTINGS

5.1. Dataset and Evaluation Metric

We considered the DESED dataset and our proposed WildDESED dataset, featuring 10-second audio clips across various subsets. All clips were resampled to 16 kHz mono and segmented using a 2048-sample window and 256-sample hop length for spectrogram extraction and log-mel spectrogram generation. Our systems were evaluated using the threshold-independent polyphonic sound event detection scores (PSDS) [36] in two scenarios following DCASE 2023 Challenge Task 4A protocol. Scenario-1 focuses on prompt reaction and temporal localization, while Scenario-2 emphasizes on reducing class confusion for SED.

5.2. Implementation Details

For our experiments, following the DCASE 2023 Task 4A baseline [29], we utilized a batch size of 48 and employed the Adam optimizer with an initial learning rate of 0.001, coupled with an exponential warmup scheduler applied across the first 50 epochs out of a total 200 epochs. To stabilize training, we implemented a mean teacher model with an exponential moving average [37] factor set at 0.999. We consider the CRNN [29] baseline system from DCASE 2023 Task 4A, featuring approximately 1.2 million parameters, ensuring a robust comparison for our curriculum learning approach.

6. RESULTS AND DISCUSSION

Table 2 shows the results of our studies on DESED and newly created WildDESED datasets. It is observed that the performance of the baseline CRNN model trained using DESED dataset drops significantly as the noise levels are increased on WildDESED dataset compared to that on the original DESED dataset. We then explore the baseline CRNN model trained using WildDESED data, which we refer to as CRNN (W). We find that CRNN (W) performs better than the original CRNN model when the noise levels on WildDESED are on the higher end (0 dB and -5 dB). However, the performance is comparable for both models when noise level is 5 dB and then the original CRNN model performs better for less noisy scenario of 10dB on WildDESED and on the clean DESED dataset.

We now focus on the studies for curriculum learning approach applied on the CRNN model trained using WildDESED dataset. We refer this model as CRNN (W+CL) and find that it outperforms both CRNN as well as CRNN (W) models for all noise levels on the WildDESED dataset. This highlights the scope of curriculum learning approach for developing noise-robust SED systems using WildDESED dataset for unseen complex domestic settings. We also note that the CRNN model trained on the clean DESED performs the best on the DESED test due to the matched conditions. However, the model CRNN (W+CL) with curriculum learning certainly helps to boost the performance of the CRNN (W) model trained on WildDESED dataset to bring it closer that of the CRNN model on DESED test set. The future work will focus on reducing this performance gap on the clean scenario for noise-robust SED models.

7. CONCLUSION

In this work, we have presented a new dataset referred to as WildDESED to advance SED research under noisy home settings and also explored a preliminary curriculum learning method to develop noise-robust SED systems. We used 12 noises from Audioset to craft the WildDESED dataset considering 8 different scenarios depicting complex home environments by considering assistance from an LLM. The studies conducted showed the scope of curriculum learning approach for developing noise-robust SED systems using the WildDESED dataset. We believe this WildDESED dataset will be useful for future horizons of noise-robust SED research.

8. REFERENCES

[1] A. Mesaros, T. Heittola, T. Virtanen, and M. D. Plumbley, "Sound Event Detection: A Tutorial," *IEEE Signal Processing Magazine*, vol. 38, no. 5, pp. 67–83, 2021.

[2] T. Khandelwal, R. K. Das, and E. S. Chng, "Sound Event Detection: A Journey Through DCASE Challenge Series," *APSIPA Transactions on Signal and Information Processing*, vol. 13, 2024.

[3] R. Radhakrishnan, A. Divakaran, and A. Smaragdis, "Audio Analysis for Surveillance Applications," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics.*, 2005, pp. 158–161.

[4] J. P. Bello, C. Silva, O. Nov, R. L. Dubois, A. Arora, J. Salamon, C. Mydlarz, and H. Doraiswamy, "SONYC: A System for Monitoring, Analyzing, and Mitigating Urban Noise Pollution," *Communications of the ACM*, vol. 62, no. 2, pp. 68–77, 2019.

[5] Y. Xiao and R. K. Das, "Dual Knowledge Distillation for Efficient Sound Event Detection," *arXiv:2402.02781*, 2024.

[6] Y. Xiao, T. Khandelwal, and R. K. Das, "FMSG Submission for DCASE 2023 Challenge Task 4 on Sound Event Detection with Weak Labels and Synthetic Soundscapes," DCASE 2023 Challenge, Tech. Rep., 2023.

[7] F. Ronchini and R. Serizel, "Performance and Energy Balance: A Comprehensive Study of State-of-the-art Sound Event Detection Systems," *arXiv:2310.03455*, 2023.

[8] H. Dinkel, M. Wu, and K. Yu, "Towards Duration Robust Weakly Supervised Sound Event Detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 887–900, 2021.

[9] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio Set: An Ontology and Human-Labeled Dataset for Audio Events," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 776–780.

[10] P. Zinemanas, P. Cancela, and M. Rocamora, "MAVD: A Dataset for Sound Event Detection in Urban Environments." in *Proc. the Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, 2019.

[11] N. Turpault, R. Serizel, A. P. Shah, and J. Salamon, "Sound Event Detection in Domestic Environments with Weakly Labeled Data and Soundscape Synthesis," in *Proc. the Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, 2019.

[12] N. Turpault and R. Serizel, "Training Sound Event Detection on a Heterogeneous Dataset," in *Proc. the Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, 2020.

[13] Y. Choi, O. Atif, J. Lee, D. Park, and Y. Chung, "Noise-robust Sound-event Classification System with Texture Analysis," *Symmetry*, vol. 10, no. 9, p. 402, 2018.

[14] I. McLoughlin, H. Zhang, Z. Xie, Y. Song, and W. Xiao, "Robust Sound Event Classification Using Deep Neural Networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 540–552, 2015.

[15] M. Neri, F. Battisti, A. Neri, and M. Carli, "Sound Event Detection for Human Safety and Security in Noisy Environments," *IEEE Access*, vol. 10, pp. 134 230–134 240, 2022.

[16] R. Serizel, N. Turpault, A. Shah, and J. Salamon, "Sound Event Detection in Synthetic Domestic Environments," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 86–90.

[17] T. Wan, Y. Zhou, Y. Ma, and H. Liu, "Noise Robust Sound Event Detection Using Deep Learning and Audio Enhancement," in *Proc. IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, 2019, pp. 1–5.

[18] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language Models are Few-shot Learners," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 1877–1901.

[19] "(toolqa: A dataset for llm question answering with external tools."

[20] Y. Gong, H. Luo, A. H. Liu, L. Karlinsky, and J. R. Glass, "Listen, Think, and Understand," in *Proc. International Conference on Learning Representations (ICLR)*, 2023.

[21] J. Bai, H. Yin, M. Wang, D. Shi, W.-S. Gan, J. Chen, and S. Rahardja, "AudioLog: LLMs-Powered Long Audio Logging with Hybrid Token-Semantic Contrastive Learning," *arXiv preprint:2311.12371*, 2023.

[22] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum Learning," in *Proc. Annual International Conference on Machine Learning (ICML)*, 2009, pp. 41–48.

[23] X. Wang, Y. Chen, and W. Zhu, "A Survey on Curriculum Learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 9, pp. 4555–4576, 2021.

[24] A. Pentina, V. Sharmanska, and C. H. Lampert, "Curriculum Learning of Multiple Tasks," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 5492–5500.

[25] S. Braun, D. Neil, and S.-C. Liu, "A Curriculum Learning Method for Improved Noise Robustness in Automatic Speech Recognition," in *Proc. European Signal Processing Conference (EUSIPCO)*, 2017, pp. 548–552.

[26] D. Ng, Y. Xiao, J. Q. Yip, Z. Yang, B. Tian, Q. Fu, E. S. Chng, and B. Ma, "Small Footprint Multi-channel Network for Keyword Spotting with Centroid Based Awareness," in *Proc. INTERSPEECH*, 2023, pp. 296–300.

[27] Y. Xiao and R. K. Das, "UCIL: An Unsupervised Class Incremental Learning Approach for Sound Event Detection," *arXiv:2407.03657*, 2024.

[28] E. Çakir, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional Recurrent Neural Networks for Polyphonic Sound Event Detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1291–1303, 2017.

[29] F. Ronchini, S. Cornell, R. Serizel, N. Turpault, E. Fonseca, and D. P. W. Ellis, "Description and Analysis of Novelties Introduced in DCASE Task 4 2022 on the Baseline System," in *Proc. the Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, 2022.

[30] T. Khandelwal and R. K. Das, "A Multi-Task Learning Framework for Sound Event Detection using High-level Acoustic Characteristics of Sounds," in *Proc. INTERSPEECH*, 2023, pp. 1214–1218.

[31] F. Ronchini, R. Serizel, N. Turpault, and S. Cornell, "The Impact of Non-Target Events in Synthetic Soundscapes for Sound Event Detection," in *Proc. the Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, 2021, pp. 115–119.

[32] G. Dekkers, S. Lauwereins, B. Thoen, M. W. Adhana, H. Brouckxon, B. Van den Bergh, T. Van Waterschoot, B. Vanrumste, M. Verhelst, and P. Karsmakers, "The sins database for detection of daily activities in a home environment using an acoustic sensor network," in *Proc. the Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, 2017, pp. 1–5.

[33] A. Mesaros, T. Heittola, and T. Virtanen, "Tut database for acoustic scene classification and sound event detection," in *Proc. European Signal Processing Conference (EUSIPCO)*. IEEE, 2016, pp. 1128–1132.

[34] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint:1510.08484*, 2015.

[35] L. Prechelt, "Early Stopping-but When?" *Neural Networks: Tricks of the trade*, pp. 55–69, 2002.

[36] J. Ebbers, R. Haeb-Umbach, and R. Serizel, "Post-Processing Independent Evaluation of Sound Event Detection Systems," in *Proc. the Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, 2023.

[37] A. Tarvainen and H. Valpola, "Mean Teachers are Better Role Models: Weight-averaged Consistency Targets Improve Semi-supervised Deep Learning Results," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.

INTEGRATING CONTINUOUS AND BINARY RELEVANCES IN AUDIO-TEXT RELEVANCE LEARNING

Huang Xie, Khazar Khorrami, Okko Räsänen, Tuomas Virtanen

Signal Processing Research Center, Tampere University, Finland

ABSTRACT

Audio-text relevance learning refers to learning the shared semantic properties of audio samples and textual descriptions. The standard approach uses binary relevances derived from pairs of audio samples and their human-provided captions, categorizing each pair as either positive or negative. This may result in suboptimal systems due to varying levels of relevance between audio samples and captions. In contrast, a recent study used human-assigned relevance ratings, i.e., continuous relevances, for these pairs but did not obtain performance gains in audio-text relevance learning. This work introduces a relevance learning method that utilizes both human-assigned continuous relevance ratings and binary relevances using a combination of a listwise ranking objective and a contrastive learning objective. Experimental results demonstrate the effectiveness of the proposed method, showing improvements in language-based audio retrieval, a downstream task in audio-text relevance learning. In addition, we analyze how properties of the captions or audio clips contribute to the continuous audio-text relevances provided by humans or learned by the machine.

Index Terms— Audio-text learning, continuous relevance, binary relevance, contrastive learning, learn-to-rank

1. INTRODUCTION

Audio-text relevance learning refers to learning the shared semantic properties of audio samples and textual descriptions. It plays an important role in applications such as language-based audio retrieval [1]. Recent studies [2, 3] address this problem with similarity learning approaches, which learn intermediate representations of audio samples and texts in a shared embedding space, thereby measuring audio-text relevance by employing a similarity function (e.g., cosine similarity) over these representations.

Audio-caption datasets (e.g., Clotho [4], WavCaps [5]), which consist of audio samples accompanied by human annotated captions, are widely used for training [1]. Due to the lack of relevance information about audio samples and captions beyond the annotated ones, a binary relevance is adopted between audio samples and captions. An audio sample is considered relevant to its annotated caption but irrelevant to all other captions. By optimizing a contrastive learning objective (e.g., InfoNCE [6]), a learning system is trained to project audio samples and their relevant captions close to each other but far away from the irrelevant ones in the shared embedding space.

It is likely that audio samples and captions can have varying levels of relevance, ranging from fully relevant to partially relevant. For instance, consider an audio sample and its corresponding caption “people speak to each other and a cat sighs”, as shown in Fig. 1. A partial caption “people speak to each other” only describes part of the audio sample, as it lacks a description of cat sighs. Therefore, it can be seen as partially relevant to the audio sample. However, when adopting binary audio-caption relevance, all captions except the one corresponding to the audio sample are regarded as irrelevant. In the given example, the caption “people speak to each other” will be incorrectly regarded as irrelevant to the audio sample. To accurately depict the relevance between audio samples and captions, it is essential to employ non-binary relevance measures (e.g., graded relevance [7, 8]).

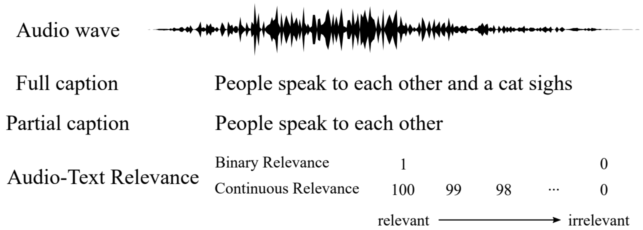


Figure 1: An audio sample with its full caption “people speak to each other and a cat sighs”, which provides a complete description of its content, and a partial caption “people speak to each other”, which only describes part of its content. The figure illustrates two categories of audio-text relevance: binary and continuous.

Current audio-caption datasets [4, 5] lack annotated non-binary relevance information for their audio samples and captions. Our previous study [9] collected continuous audio-caption relevances for a small subset of Clotho [4] via crowdsourced subjective assessments. Human annotators were asked to assign relevance ratings (ranging from 0 to 100) to audio samples with respect to a given caption. It was shown that reducing these ratings to binary relevances for training did not improve model performance on downstream tasks (e.g., language-based audio retrieval) [9]. Conversely, obtaining continuous relevances through subjective assessments is often expensive, being labor-intensive and time-consuming, while training a system typically requires a large amount of data.

This work proposes an audio-text relevance learning method that leverages both continuous and binary relevances. We train modality-specific encoders to project audio samples and texts into a shared embedding space, learning audio-text relevance by computing the cosine similarity of their embeddings. During training, we jointly optimize a listwise ranking objective (e.g., ListNet [10]) with human-assigned continuous relevance ratings and a contrastive learning objective (e.g., InfoNCE [6]) with binary audio-text relevances. Experimental results demonstrate the effectiveness of the proposed method, showing improvements in language-based audio retrieval, a downstream task in audio-text relevance learning. Additionally, we analyze how properties of the captions or audio clips contribute to the continuous audio-text relevances provided by humans or learned by the machine.

2. PROPOSED METHOD

This section presents the proposed audio-text relevance learning method with continuous and binary relevances.

2.1. Audio-Text Relevance Learning

Fig. 2 presents a model-agnostic dual-encoder framework for audio-text relevance learning. Audio samples and texts are projected into a shared embedding space via modality-specific encoders (e.g., audio and text encoders). The relevance between audio samples and texts is measured by computing the similarity between their embeddings, such as cosine similarity. When training with continuous relevances (e.g., relevance ratings), a listwise ranking objective (e.g., ListNet [10]) is computed based on these relevances and the audio-text embedding similarities. For binary relevances, a contrastive learning objective (e.g., InfoNCE [6]) is calculated from the binary relevances and embedding similarities. Using both relevances for training involves a joint learning objective combining the listwise ranking and contrastive learning objectives.

2.2. Learning Objectives

ListNet Loss with Continuous Relevances. Given N audio samples $\{y_1, \cdots, y_N\}$ and a caption x, let s_i $(0 \leq s_i \leq 100)$ be the relevance rating of y_i to x, for $i = 1, \cdots, N$. Suppose that L is an ideal ranked list (which is unknown), where $y_1, \cdots, y_N$ are arranged in descending order according to their relevance to x. The top-one ranking probability of y_i, denoted by $p(y_i)$, represents the probability of it being ranked at the top of L, given the relevance ratings $\{s_1, \cdots, s_N\}$.

Inspired by [10], $p(y_i)$ is written as

$$p(y_i) = \frac{\phi(s_i)}{\sum_{j=1}^{N} \phi(s_j)}, \tag{1}$$

with $\phi(s_i)$ being

$$\phi(s_i) = \frac{s_i}{\log_2 \left(r(s_i) + 1\right)}, \tag{2}$$

where $r(s_i)$ represents the position of s_i in the descending ranked list of $s_1, \cdots, s_N$. The top-one probabilities $\{p(y_1), \cdots, p(y_N)\}$ define a probability distribution P over audio samples $y_1, \cdots, y_N$ with respect to the caption x.

In this work, the dual-encoder framework outputs cosine similarity scores between audio and text embeddings as a measure of audio-text relevance. Suppose that $\mathbf{a}_i$ is the audio embedding of y_i, and $\mathbf{c}$ is the text embedding of x; their similarity score is denoted by $t(\mathbf{a}_i, \mathbf{c})$, with $t(\mathbf{a}_i, \mathbf{c}) \in [-1, 1]$.

Similar to (1), we calculate another top-one ranking probability of y_i, denoted by $q(y_i)$, based on the similarity scores $\{t(\mathbf{a}_1, \mathbf{c}), \cdots, t(\mathbf{a}_N, \mathbf{c})\}$. Specifically, the probability $q(y_i)$ is written as

$$q(y_i) = \frac{\exp(t(\mathbf{a}_i, \mathbf{c})/\omega)}{\sum_{j=1}^{N} \exp(t(\mathbf{a}_j, \mathbf{c})/\omega)}. \tag{3}$$

with ω being a hyperparameter. These top-one probabilities $\{q(y_1), \cdots, q(y_N)\}$ define another probability distribution Q over audio samples $y_1, \cdots, y_N$ with respect to the caption x.

Finally, the ListNet loss is calculated as the cross entropy between the two probability distributions P and Q, written as

$$L_{\text{ListNet}} = -\sum_{j=1}^{N} p(y_j) \log q(y_j). \tag{4}$$

By minimizing (4), the dual-encoder framework are optimized for ranking audio samples by their relevance to a given caption.

InfoNCE Loss with Binary Relevances. For the case of binary relevances, InfoNCE loss [6] is used for training. In InfoNCE, audio samples and captions are considered relevant only if they correspond to each other, and otherwise, they are irrelevant. Similar to [2, 3], we use symmetric InfoNCE that tries to classify audio samples as either relevant or irrelevant to a given caption, and vice versa. The total InfoNCE loss is then calculated as the sum of two categorical cross entropies of the two tasks.

Dataset	Relevance	#Audio / #Captions		
		development	**validation**	**evaluation**
GrRel	Graded	2186 / 200	1004 / 200	1009 / 200
BiRel	Binary	2186 / 200	1004 / 200	1009 / 200
SuperBiRel		2186 / 2186	1004 / 1004	1009 / 1009
Clotho		3839 / 19195	1045 / 5225	1045 / 5225

Table 1: Audio samples and captions with human-assigned relevance ratings (i.e., graded relevances) and binary relevances.

Joint Loss. To use both continuous and binary audio-text relevances for training, the ListNet (4) and InfoNCE losses are combined into a joint loss. Specifically, the joint loss is written as

$$L_{\text{joint}} = \alpha \cdot L_{\text{InfoNCE}} + (1 - \alpha) \cdot L_{\text{ListNet}}, \tag{5}$$

where α is a hyperparameter that is chosen from $(0, 1)$.

3. EXPERIMENTS

The proposed method was validated on language-based audio retrieval, a downstream task in audio-text relevance learning.

3.1. Audio and Text Data

Clotho. All audio samples and texts used in the experiment were selected from Clotho [4], which consists of 5,929 audio samples, each with five human annotated captions. Clotho is partitioned into three subsets: a development set with 3,839 audio samples, a validation set with 1,045 audio samples, and an evaluation set with 1,045 audio samples.

Continuous Relevances. Our previous study [9] collected relevance ratings for a small subset of audio samples and captions in Clotho [4] via crowdsourced subjective assessments. Human annotators were asked to assign relevance ratings (ranging from 0 to 100) to indicate their judgements of how much the acoustic content of an audio sample matched with a given caption. Relevance ratings were collected for 17 audio samples per caption across 600 captions, resulting in a total of 10,200 ratings. We denoted this subset of Clotho with human-assigned relevance ratings as “GrRel”.

Binary Relevances. We constructed three datasets of audio samples and captions with binary relevances, where an audio sample was considered relevant to its corresponding caption but irrelevant to all other captions in the dataset. Specifically, we denoted as “BiRel” the set of audio samples and captions, the same as “GrRel”, but annotated with binary relevances. The “SuperBiRel” consisted of audio samples in “BiRel” and “GrRel”, each accompanied by one of the five reference captions provided in Clotho [4]. Finally, all audio samples and captions in Clotho [4] were utilized for experiment. Regardless of the graded and binary relevances in these datasets, we had “GrRel” = “BiRel” $\subset$ “SuperBiRel” $\subset$ Clotho. Table 1 summarizes the four datasets.

3.2. Audio and Text Encoders

Audio Encoder. A pretrained CNN14 [11] was utilized as the audio encoder, with a fully-connected layer added on its top. It took 64-dimensional log mel-band energies as inputs, which were computed

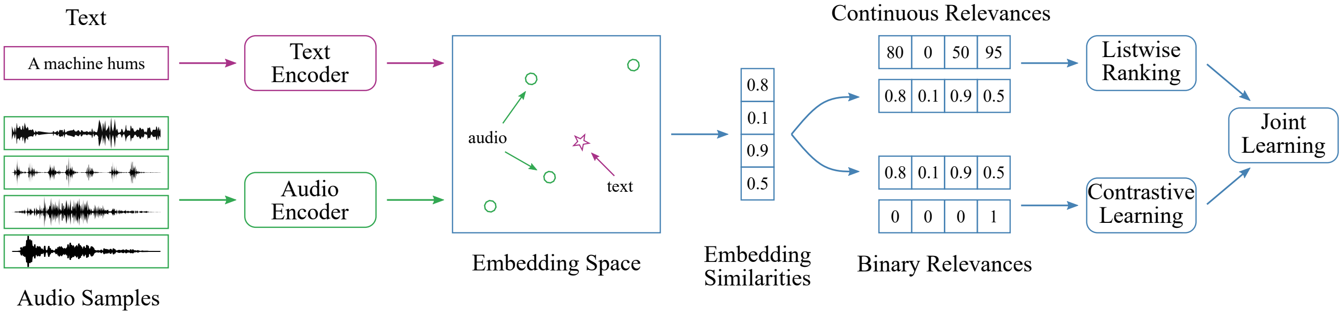


Figure 2: A model-agnostic dual-encoder framework for audio-text relevance learning with continuous and binary relevances.

from 40 ms Hanning-windowed frames with a hop length of 20 ms, and generated 300-dimensional audio embeddings. We fine-tuned the audio encoder during training.

Text Encoder. The Sentence-BERT (specifically, “all-mpnet-base-v2”) [12] was employed as the text encoder to extract 768-dimensional text embeddings from audio captions. An additional fully-connected layer was added on top to transform these text embeddings into 300-dimensional embeddings. The Sentence-BERT was frozen during training.

Training Setup. During training, optimization was carried out using the Adam optimizer, starting with a learning rate of 0.001. If the validation loss failed to improve over five consecutive epochs, the learning rate was reduced by a factor of ten. Early stopping was applied with a patience of ten epochs to terminate training if no improvement was observed.

3.3. Language-based Audio Retrieval

The proposed method was validated on language-based audio retrieval, a downstream task of audio-text relevance learning, which aims to retrieve audio samples from a dataset based on their relevance to a given textual query. We performed language-based audio retrieval by using captions as textual queries to retrieve their corresponding audio samples in the Clotho evaluation set (“Clotho-evaluation”) [4]. Audio-text relevance was measured by the cosine similarity between audio and text embeddings, with higher cosine similarity indicating greater relevance.

Evaluation Metrics. Retrieval performance was assessed using mean Average Precision (mAP) and Recall at 10 (R@10), as done in [1]. The mAP was determined by averaging the Average Precision (AP) scores across all query captions, where AP was the average of the precision values at the positions of audio samples in the relevance-based ranked list corresponding to a query caption. R@10 was calculated as the proportion of audio samples within the top-10 results relative to the total number of audio samples corresponding to a query caption, averaged over all captions. Higher values for both metrics indicate better performance.

4. RESULTS

This section reports the experimental results for language-based audio retrieval and audio-text relevance learning.

4.1. Language-based Audio Retrieval

Table 2 presents the results on Clotho-evaluation. Each evaluation is repeated five times, and the averaged metrics are reported.

Training Dataset	**Loss**	**Evaluation Metrics**	
		mAP	**R@10**
GrRel	ListNet	0.034 ± 0.001	0.070 ± 0.002
BiRel		0.015 ± 0.002	0.024 ± 0.004
SuperBiRel	InfoNCE	0.168 ± 0.005	0.356 ± 0.010
Clotho		0.239 ± 0.001	0.482 ± 0.002
SuperBiRel + GrRel	Joint	0.173 ± 0.001	0.364 ± 0.009
Clotho + GrRel		0.244 ± 0.002	0.486 ± 0.002

Table 2: Language-based audio retrieval on Clotho-evaluation.

Continuous and Binary Relevances. Note that the ListNet loss (see Section 2.2) can also work with binary relevances. In such a case, the binary relevance values $\{0, 1\}$ are mapped to relevance ratings $\{0, 100\}$, respectively. When training the dual-encoder framework with the ListNet loss (4), we experimented with the same audio samples and captions with either human-assigned relevance ratings (“GrRel”) or binary relevances (“BiRel”). Experimental results show that using human-assigned relevance ratings for training leads to better performance. For instance, “GrRel” surpasses “BiRel”, achieving an R@10 score of 0.070 ± 0.002 compared to 0.024 ± 0.004 for “BiRel”. We conclude that continuous relevances (e.g., human-assigned relevance ratings) outperform binary relevances in depicting the relevance between audio samples and texts, thereby resulting in superior performance in language-based audio retrieval.

Learning Objectives. When training the dual-encoder framework with different learning objectives, the joint loss (5) achieves enhanced performance compared to both the ListNet loss (4) and the InfoNCE loss [6]. For instance, the InfoNCE loss with “SuperBiRel” yields an R@10 score of 0.356 ± 0.010, and the ListNet loss with “GrRel” obtains an R@10 score of 0.070 ± 0.002. When combined, the joint loss with “GrRel” and “SuperBiRel” attains an R@10 score of 0.364 ± 0.009. This demonstrates the effectiveness of the proposed method in utilizing both human-assigned relevance ratings and binary relevances for audio-text relevance learning.

Additionally, regardless of the learning objectives, the volume of training data (e.g., the number of audio samples and captions) affects performance. For instance, when working with the InfoNCE loss, the larger Clotho outperforms “SuperBiRel”, achieving an R@10 score of 0.482 ± 0.002 compared to 0.356 ± 0.010 for “SuperBiRel”. The ListNet loss obtains the worst performance, likely due to the limited number of audio samples and captions in “GrRel” and “BiRel”.

4.2. Learned Audio-Text Relevances

The learned audio-text relevances are measured by the cosine similarity between audio and text embeddings, with higher similarity

Training Dataset	**Loss**	**Correlation**	
		ρ**-statistic**	**p-value**
GrRel	ListNet	0.530	< 0.001
SuperBiRel	InfoNCE	0.671	< 0.001
SuperBiRel + GrRel	Joint	0.690	< 0.001

Table 3: Spearman's rank-order correlation between learned relevances and human-assigned ratings in the "GrRel" evaluation set.

	Feature	HR	MR	D(H, M)	APT
Audio	e-time	n.s.	n.s.	n.s.	n.s.
	e-class	-0.195**	-0.233**	n.s.	0.089*
	audio duration	n.s.	n.s.	n.s.	0.604**
Text	perplexity	-0.092*	n.s.	n.s.	n.s.
	# words	0.130**	n.s.	0.108**	0.227**
	# C-words	0.099*	n.s.	0.100*	0.213**
	# nouns	n.s.	n.s.	0.107**	0.158**
	# adjectives	n.s.	n.s.	n.s.	0.082*
	# fr-words	0.095*	n.s.	n.s.	0.113**
	# fr-C-words	0.089*	n.s.	n.s.	0.127**
	# fr-nouns	n.s.	n.s.	0.083*	0.095*

Table 4: Pearson correlation coefficients between annotation characteristics and data features. * p-value <0.05, ** p-value <0.01, ”n.s.” = not significant.

indicating greater relevance. We collect learned audio-text relevances for audio samples and captions in the "GrRel" evaluation set from three setups: the ListNet loss (4) with "GrRel", the InfoNCE loss [6] with "SuperBiRel", and the joint loss (5) with both datasets. These learned relevances are compared with human-assigned relevance ratings in the "GrRel" evaluation set, which includes 3,400 relevance ratings across 1,009 audio samples and 200 captions. Specifically, we calculate Spearman's rank-order correlation [13] between the learned audio-text relevances and human-assigned relevance ratings.

As shown in Table 3, all three setups learn audio-text relevances that are moderately positively correlated with the human-assigned relevance ratings ($0.4 < \rho < 0.7$, p-values < 0.001). Employing the joint loss with both datasets yields the highest correlation observed with the human-assigned relevance ratings ($\rho = 0.690$, p-value < 0.001). Despite the ListNet loss applied to "GrRel", which incorporates the fewest audio samples and captions during training, it demonstrates a moderate positive correlation ($\rho = 0.53$, p-value < 0.001), showing the effectiveness of the proposed method in audio-text relevance learning.

5. ANALYSIS OF CROWDSOURCED RATINGS

To better understand the data-related bias factors in our audio-text relevance annotations and learning systems, we analyzed how text and audio properties might be related to the relevance ratings provided by humans and learned by the machine. Previous studies in crowd-sourcing sound events have linked annotation characteristics like inter-annotator agreement to audio attributes such as overlapping sounds [14] and sound-event loudness [15]. Regarding data attributes, research has explored audio complexity through spectral dynamics [16] and cognitive processing demands [17], as well as text complexity based on caption length and syntactic structure [18].

Inspired by these works, we measured a set of audio and text features and analyzed their correlation with human and machine audio-caption relevance ratings. We conducted our analysis across true-positive (TP) pairs (i.e., 600 crowdsourced captions and their original audio pairs [4], see [9]). For audio clips, we utilized the probability matrix from the pre-trained sound event detection PANNS model [11], measuring entropy across 527 sound classes (e-class) using averaged probabilities across time, and entropy over time (e-time) using averaged probabilities across classes, where entropy $H = -\sum_i p_i \log_2(p_i)$, with p_i as probability distributions. Additionally, we considered the audio clip duration as a third attribute. Regarding text captions, we analyzed perplexity (as a measure of syntactic complexity), word count, content words (C-words) count, and adjectives count. Additionally, we compiled lists of the 500 most frequent (fr) words, content words, and adjectives from the original Clotho dataset, tallying their occurrences in each caption.

Audio and text attributes were correlated against the human-rated (HR) audio-text relevances and their standard deviation across annotators (SD-HR), the latter being a measure of annotators' disagreement on the relevances. The attributes were also compared against machine relevance ratings (MR) from a model trained on Clotho data (InfoNCE loss with binary relevances) to understand the factors contributing to the learnability of the data. Finally, we explored whether the audio/text attributes can explain the degree of disagreement between HR and MR (D(H, M)), and whether the average time annotators spent playing audio clips (APT) correlates with audio/text attributes.

Table 4 presents the Pearson correlation coefficients between the measured audio and text features and the annotation characteristics. HR and MR are negatively correlated with class entropy, indicating that the presence of diverse sound classes in the clip leads both humans and machines to score a TP audio-text pair as less relevant. Conversely, a positive correlation (r=0.213, p<0.01) between std-HR and class entropy suggests that such a feature results in disagreement among annotators. Moreover, the syntactic complexity of captions tends to lead annotators to score a true positive audio-caption pair as less relevant. Conversely, longer and denser captions tend to lead annotators to perceive audio-caption pairs as more relevant. Similarly, the disagreement between human and machine ratings increases with longer and denser captions. As expected, annotators tended to play audio clips for a longer duration when the actual length of the audio was longer. Surprisingly, the average played time is also correlated with text attributes, suggesting that when annotators are presented with a longer and denser caption, they tend to listen to more of the audio clip before rating the relevance of the audio-caption pair.

6. CONCLUSIONS

This study introduced an approach to audio-text relevance learning that integrated both continuous and binary relevances. By training modality-specific encoders, we projected audio samples and texts into a shared embedding space, where the cosine similarity of their embeddings served as a measure of their relevance. Through a combined optimization of a listwise ranking objective using continuous relevance ratings and a contrastive learning objective with binary relevances during training, our method demonstrated enhanced performance in language-based audio retrieval, a downstream task in this domain. Moreover, we analyzed how various properties of captions and audio clips influenced both human-assigned and machine-learned continuous relevances.

7. REFERENCES

[1] H. Xie, S. Lipping, and T. Virtanen, "Language-based audio retrieval task in dcase 2022 challenge," in *Proc. Detect. Classif. Acoust. Scenes Events Work. (DCASE)*, 2022, pp. 216–220.

[2] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, "Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation," in *Proc. Int. Conf. Acoustic., Speech and Signal Process. (ICASSP)*, 2023, pp. 1–5.

[3] P. Primus, K. Koutini, and G. Widmer, "Advancing natural-language based audio retrieval with passt and large audio-caption data sets," in *Proc. Detect. Classif. Acoust. Scenes Events Work. (DCASE)*, 2023, pp. 151–155.

[4] K. Drossos, S. Lipping, and T. Virtanen, "Clotho: An audio captioning dataset," in *Proc. Int. Conf. Acoustic., Speech and Signal Process. (ICASSP)*, 2020, pp. 736–740.

[5] X. Mei, C. Meng, H. Liu, Q. Kong, T. Ko, C. Zhao, M. D. Plumbley, Y. Zou, and W. Wang, "Wavcaps: A chatgpt-assisted weakly-labelled audio captioning dataset for audio-language multimodal research," *IEEE/ACM Trans. Audio Speech Lang. Process.*, pp. 1–15, 2024.

[6] A. Van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, arXiv preprint arXiv:1807.03748.

[7] T. Sakai, "Graded relevance," in *Evaluating Information Retrieval and Access Tasks: NTCIR's Legacy of Research Impact*. Singapore: Springer Singapore, 2021, pp. 1–20.

[8] K. Roitero, E. Maddalena, S. Mizzaro, and F. Scholer, "On the effect of relevance scales in crowdsourcing relevance assessments for information retrieval evaluation," *Inf. Process. Manag.*, vol. 58, no. 6, p. 102688, 2021.

[9] H. Xie, K. Khorrami, O. Räsänen, and T. Virtanen, "Crowdsourcing and evaluating text-based audio retrieval relevances," in *Proc. Detect. Classif. Acoust. Scenes Events Work. (DCASE)*, 2023, pp. 226–230.

[10] Z. Cao, T. Qin, T. Liu, M. Tsai, and H. Li, "Learning to rank: from pairwise approach to listwise approach," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2007, pp. 129–136.

[11] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. Plumbley, "Panns: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Trans. Audio Speech Lang. Process.*, pp. 2880–2894, 2020.

[12] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using siamese BERT-networks," in *Proc. Empherical Methods Nat. Lang. Process. (EMNLP)*, 2019, pp. 3982–3992.

[13] C. Spearman, "The proof and measurement of association between two things," *Am. J. Psychol.*, vol. 15, no. 1, pp. 72–101, 1904.

[14] I. Martín-Morató and A. Mesaros, "Strong labeling of sound events using crowdsourced weak labels and annotator competence estimation," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 31, pp. 902–914, 2023.

[15] M. Cartwright, J. Salamon, A. Seals, O. Nov, and J. P. Bello, "Investigating the effect of sound-event loudness on crowdsourced audio annotations," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2018, pp. 341–345.

[16] N. C. Singh, "Measuring the 'complexity'of sound," *Pramana*, vol. 77, pp. 811–816, 2011.

[17] A. Lang, Y. Gao, R. F. Potter, S. Lee, B. Park, and R. L. Bailey, "Conceptualizing audio message complexity as available processing resources," *Commun. Res.*, vol. 42, no. 6, pp. 759–778, 2015.

[18] D. Brunato, L. De Mattei, F. Dell'Orletta, B. Iavarone, G. Venturi, *et al.*, "Is this sentence difficult? do you agree?" in *Proc. Empir. Methods Nat. Lang. Process. (EMNLP)*, 2018, pp. 2690–2699.

MOFLENET: A LOW COMPLEXITY MODEL FOR ACOUSTIC SCENE CLASSIFICATION

Oo Yifei[1*†] *Nagisetty Srikanth*[2†] *Chong Soon Lim*[2]

[1]Nanyang Technological University, Singapore {yoo001@e.ntu.edu.sg}
[2]Panasonic R&D Center Singapore
{srikanth.nagisetty@sg.panasonic.com, chongsoon.lim@sg.panasonic.com}

ABSTRACT

Designing lightweight models that require minimal computational resources and can operate on edge devices is the latest trend in deep learning research. This paper details our approach to Task 1: Low-Complexity Acoustic Scene Classification (ASC) for the DCASE'24 challenge. The task involves developing data-efficient systems for five scenarios, which progressively limit the available training data (i.e., 100%, 50%, 25%, 10%, 5%), while also handling device mismatches and low-complexity constraints (maximum memory allowance for model parameters: 128 kB, maximum number of MACs per inference: 30 million). In this work, we introduce a lightweight novel CNN architecture called MofleNet, featuring a combination of shuffle channels and residual inverted bottleneck blocks. Furthermore, we improve the performance by ensembling MofleNet with CP-ResNet. To meet the constraint of keeping the model size under 128 kB, both models are fine-tuned using quantization-aware training. Compared to the DCASE'24 Task 1 baseline, our proposed system improves results on the TAU Urban Acoustic Scenes 2022 Mobile Development dataset by around 6% on an average across five datasets and 4% on the challenge test set, earning a 7th rank in the DCASE'24 task 1 challenge.

Index Terms— MofleNet, CP-ResNet, Ensemble Learning, Quantization Aware Training, Device Impulse Response augmentation, Freq-MixStyle

1. INTRODUCTION

Acoustic Scene Classification (ASC) is a key research area within computational auditory scene analysis, focusing on categorizing audio recordings into predefined scene types. ASC has the potential to enhance various applications, including wearable devices, robotics, smart home devices, autonomous vehicles, and environmental monitoring. The annual *IEEE* DCASE Challenge has driven significant progress in ASC over the years.

In the *IEEE* DCASE'24 Challenge Task 1 [1], the goal is to classify 1-second audio recordings into one of ten predefined acoustic scene classes under three challenging conditions: (1) a recording device mismatch, (2) low complexity constraints, and (3) limited training data. For the training data, five scenarios with data subsets containing data approx. 5%, 10%, 25%, 50%, and 100% are provided. A system must only be trained on the specified subset and the explicitly allowed external resources. Additionally, to ensure ASC systems perform well on typical edge devices, strict constraints are imposed, limiting model size to 128 kB and multiply-accumulate operations (MACs) per inference to 30 million.

Convolutional Neural Networks (CNNs) dominates ASC tasks. Lightweight models like MobileNet variant CP-Mobile [4], GhostNet [5], SepNet [6] and blueprint separable convolutions network [7] have been used to tackle DCASE Task 1 challenges prior to 2023. In the DCASE'23 Task 1 challenge, the rank-1 model used an ensemble of 12 teacher models, including six variants of Patchout FaSt Spectrogram Transformer (PaSST) and six variants of CP-ResNet [3], to train a student model, CP-mobile (CPM) [4]. CPM's performance depends heavily on the number of channels in each CPM Block, but reducing the model size often requires sacrificing accuracy. Moreover, scaling down the model size does not proportionately decrease the MACs, presenting a significant challenge in balancing model size, accuracy, and computational efficiency. This year's challenge requires training the system on five different sizes of training sets. Training the teacher model on a 100% dataset to distill knowledge into a student model trained on a smaller dataset is not allowed. As a result, adopting the same approach as the top-ranked submission would require training 12 models for each of the five dataset sizes, totaling to 60 teacher models, making the process highly resource-intensive.

This paper introduces MofleNet (MobileShuffleNet), a model that incorporates channel shuffling and residual inverted bottleneck blocks into the CNN network. MofleNet is efficiently designed to meet the challenge requirements and address CP-Mobile's limitations. To further improve performance, we consider an ensemble of MofleNet and optimized CP-ResNet. The remainder of the paper is structured as follows: Section 2 discuss the data preprocess. Section 3 presents the MofleNet model. Section 4 covers ensemble models. The experimental setup is covered in Section 5. Section 6 discuss results, and finally, the conclusions are presented in Section 7.

2. DATA PRE-PROCESS

2.1. Dataset

The development dataset for this challenge is TAU22 [2], containing recordings from 12 European cities and capturing 10 distinct acoustic scenes using 4 real devices. Additionally, synthetic data for 11 mobile devices was generated based on the original recordings. TAU22 retains the content of the TAU Urban Acoustic Scenes 2020 Mobile development dataset (TAU20) but segments the 10- second audio clips into 1-second fragments, significantly increasing prediction difficulty. The dataset comprises

[†] Both authors contributed equally.
[*] Work was done during an internship at Panasonic R&D Center Singapore

230,350 1-sec audio clips, each labeled with corresponding acoustic scene. All audio clips are single-channel, 44.1 kHz and 24-bit format.

2.2. Feature Extraction

Raw 1D time domain audio signals were resampled to 32 kHz and converted to Mel domain. To obtain the Mel spectrogram, time domain signal is converted to the time frequency domain using short-time Fourier transform (STFT). This ensures that both the temporal and spectral characteristics of the audio data are utilized. After the frequency domain conversion, we extracted the Mel spectrogram corresponding to each audio clip using 256 Mel bands covering upto16 kHz. For the STFT Parameters, we employ a window size of 96 ms with a hop size of 16 ms for MofleNet and 23.4 ms as hop size for CP-ResNet. Input (features extracted) is arranged in the form of Frequency Bands X Time Frames X Channels.

2.3. Data Augmentations

To mitigate overfitting, especially for limited labelled data and to achieve good generalization, combination of Freq-MixStyle, Device Impulse Responses, and time rolling techniques are used.

Frequency MixStyle (FMS) is the frequency-wise version of MixStyle. It mixes frequency-wise statistics instead of channel-wise statistics in audio processing tasks [8]. MixStyle enhances model robustness to domain shifts by normalizing input features using the mean and standard deviation of other samples within the same batch, leveraging the observation that instance-wise statistical moments encapsulate style information. FMS normalizes the frequency bands in a spectrogram and then denormalizes them with mixed frequency statistics of two spectrograms. FMS is applied to a batch with a probability specified by the hyperparameter p_{FMS}, and the mixing coefficient is drawn from a Beta distribution of parameter α.

Device Impulse Response (DIR) augmentation involves convolving the input recordings with impulse responses from 66 freely available DIRs [9] from MicIRP [10]. The characteristic frequency responses of the recording devices in MicIRP make them ideal for simulating a diverse range of recording devices. This technique is designed to enhance the model's ability to generalize across recordings from various devices. DIR augmentation is controlled by the hyperparameter p_{DIR}, which defines the probability of convolving a waveform with a DIR.

Time Rolling involves shifting a prefix/suffix of a randomly sampled length to the other end of the input signal. This augmentation, computed in the time domain, helps to simulate variations in the temporal alignment of the audio data.

Following parametric values are used for data augmentation.

Table 1: Data Augmentation Parameters

Model Name	FMS		DIR	Time Rolling
	p_{FMS}	α	p_{DIR}	
MofleNet	0.4	0.3	0.6	125ms
CP-ResNet	0.8	0.4	0.4	125ms

3. MOFLENET

Our proposed MofleNet architecture (MofleNet127) is depicted in Figure 1. It combines strided convolutions, Mofle Blocks, and average pooling to aggregate all components from the last convolution layer to obtain the scene prediction probabilities. The design of MofleNet was inspired from CP-Mobile [4] and ShuffleNet [11].

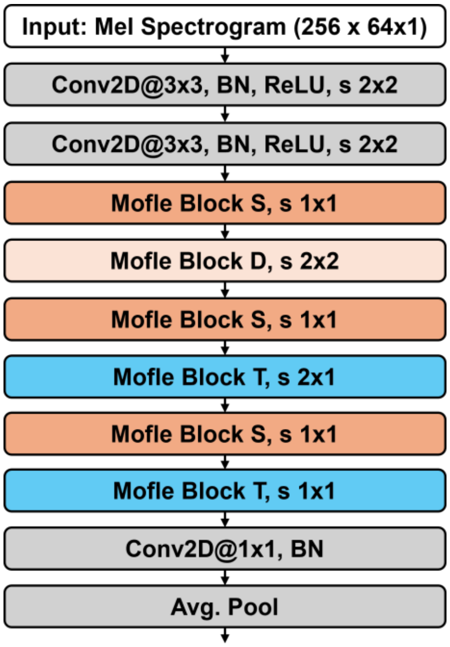


Figure 1: MofleNet127 Architecture:
Conv2D@KxK: Conv2D with Kernel Size KxK
s-Stride, BN-Batch Normalization

The Mofle block integrates grouped convolution, channel shuffle, depth wise convolution, and pointwise projection convolution to create a residual inverted bottleneck block. Figure 3 illustrates the S (Standard)/D (Spatial Down sampling) /T (Transition) design of Mofle Blocks. Unlike the CPM block, which employs pointwise expansion convolution, the Mofle Block replaces this with grouped convolution. A drawback of grouped convolution is that some channels outputs are derived from only a small fraction of input channels, limiting information exchange. To address this, channel shuffle (See Figure 2) was introduced after the grouped convolution to enhance information flow between channel groups. This promotes better mixing of information across different groups of channels, capturing more diverse and comprehensive features. This approach reduces the number of parameters and computational cost without significantly compromising information exchange between channels, resulting in richer and more informative feature maps. Additionally, the fourth layer of ShuffleNet units in [11] employs a grouped convolution, our experiments showed this configuration did not demonstrate significant improvement. Therefore, the Mofle block design doesn't include a grouped convolution layer after the depth wise convolutions.

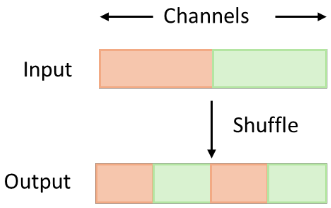


Figure 2: Channel Shuffle

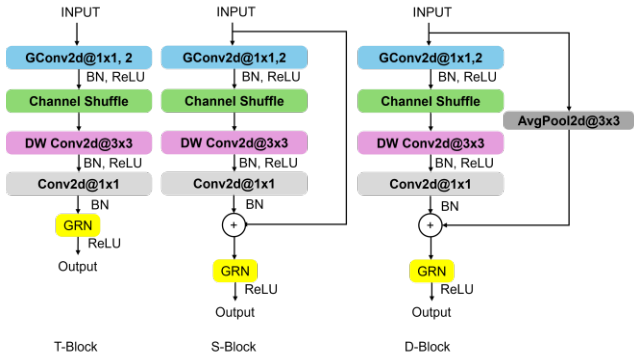


Figure 3: Mofle Blocks: T (Transition)-Block, S (Standard)-Block, D (Spatial Down sampling)-Block

T-block: The T-Block is designed to increase the number of channels within the network. These channels help in learning features across the 2D dimension, enabling the network to capture various patterns and representations from the input data.

S-block: The S-Block includes a residual connection, which helps mitigate vanishing gradient issues and facilitates the training of MofleNet. This block allows the model to learn both the original representation and the residual, leading to smoother optimization and better gradient flow.

D-block: The primary function of the D-Block is to reduce the model's complexity, particularly the MACs. It achieves this by decreasing the size of the feature maps, allowing the model to handle smaller data sizes more efficiently.

Global Response Normalization (GRN) [15] is applied before the final ReLU activation. GRN in Mofle blocks is used to avoid feature redundancies in models with restricted capacity.

4. ENSEMBLE MODELS

To enhance performance, we ensembled MofleNet and CP-ResNet after optimizing both models to meet challenge constraints. The resulting model sizes are 57kB for MofleNet127 (now referred as MofleNet57) and 59kB for CPR128 (now referred as CPR59), totaling 116kB.

4.1. MofleNet57

To lower the MACs without majorly impacting the accuracy, third Mofle Block in Figure 1 was tuned from Block S to Block D with a stride of (2x1) during convolution. Additionally, adjusted the channel multiplier and expansion rate to 1.8 and 2.6 respectively to further reduce model size and computations.

4.2. CPR59

CP-ResNet is a receptive-field regularized CNN that gradually builds local features covering a spatially restricted size. Table 2 presents the CPR59 architecture, a modified CP-ResNet, that ranked 1st in the DCASE'22 Task 1 challenge [3]. The original CP-ResNet model (CPR128) has approximately 128k parameters with a model size of 128kB. To reduce the model size and complexity, the following modifications were introduced:

- The number of parameters in the CP-ResNet network grows quadratically with its width. Reducing the channel multiplier from 2.0 to 1.4 brings down the parameter count below 64,000 (50% reduction in model size).
- Introducing max pooling layers with a shape of (2x1) and stride of (2x1) before the third and fourth (also the last) blocks reduces the MACs to under 15 million.

For additional information on the Basic CPR block listed in Table 2, refer to Figure 4.

Table 2: CPR59 Architecture

Operator	Output Shape
Input	256 x 43 x 1
Conv2D@3x3, BN, ReLU	127 x 21 x 32
Max Pool	63 x 10 x 32
Basic CPR Block(A)	63 x 10 x 32
Max Pool	31 x 10 x 32
Basic CPR Block(A)	31 x 10 x 32
Max Pool	15 x 10 x 32
Max Pool	7 x 10 x 32
Basic CPR Block(B)	7 x 10 x 44
Max Pool	7 x 5 x 44
Basic CPR Block(B)	7 x 5 x 26
Conv2D@1x1, BN	7 x 5 x 10
Avg. Pool	1 x 1 x 10

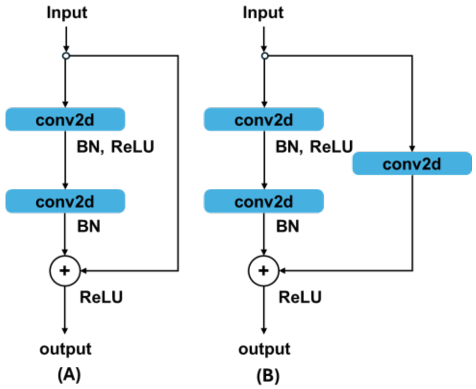


Figure 4: Two Basic CPR Blocks

5. EXPERIMENTAL SETUP

5.1. Training:

A total of 150 epochs with a batch size of 256, and adam optimizer was used for training. The learning rate strategy follows the same approach as in [4].

5.2. Quantization Aware Training:

As a part of Task 1 challenge constraints, submitted models should meet 128kB as memory requirement. To minimize the drop in performance after the quantization step, we applied Quantization Aware Training (QAT) [13] to all our architectures by fine-tuning the models for 24 epochs. A peak learning rate of 5×10−5 and linearly decreased it to 10% by epoch 16 was set during fine-tuning phase. Conv2d + BN + ReLU combinations was fused into a single layer and utilized PyTorch's 'fbgemm' quantization configuration [12]. All computations were performed in int8, except for those in the GRN layer of MofleNet. Table 3 presents the total parameters, model size and Million MACs (MMAC) per inference.

Table 5: DCASE’24 Top Submission Results

Submission Label	Rank	Accuracies per Split					Key Contribution	Knowledge Distillation
		100%	50%	25%	10%	5%		
Han_SJTUTHU	1	61.82	60.38	59.09	56.69	54.35	Model Pruning	4 Teachers models
MALACH24_JKU	2	61.51	60.05	58.01	54.46	51.95	New training strategy	3 Teacher models with Bayesian Ensemble
Shao_NEPUMSE	3	61.71	60.61	58.31	53.75	51.38	Mamba variation	12 Teacher models
OO_NTUPRDCSG	7	59.91	58.42	55.87	51.43	48.52	MofleNet	Not Utilized

Table 3: Total Parameters, Model Size and Complexity

Model	Parameters	Size (kB)	MMAC
Baseline	61,000	122	29
MofleNet127	127,000	127	27.7
MofleNet57	57,000	57	13.4
CPR59	59,000	59	16
MofleNet57+CPR59	116,000	116	29.4

6. RESULTS

6.1. Development Results

The performance of the models for the five scenarios (100%, 50%, 25%, 10%, 5%) on the validation data is shown in Table 4, the data splits are predefined by the challenge organizers. On average, the MofleNet127 and CPR128 architectures demonstrate a 4% performance improvement compared to the baseline. Notably, MofleNet127 performed well on the 100%, 50%, 25%, and 10% datasets but shows limited improvement on the 5% dataset. In contrast, CPR128 [3] outperforms MofleNet127 on the 5% dataset by 4.1%.

Table 4: Model accuracies after QAT

Model	Accuracies per Split				
	100%	50%	25%	10%	5%
Baseline	56.99	53.19	50.29	45.29	42.40
MofleNet127	61.94	58.68	55.4	49.1	42.94
CPR128	60.06	58.88	55.18	50.82	47.08
MofleNet57	58.79	56.71	52.21	45.4	41.22
CPR59	58.49	57.52	54.81	48.79	44.92
MofleNet57+ CPR59	**62.22**	**60.04**	**56.73**	**51.27**	**47.59**

Using MofleNet57 or CPR59 individually, without ensembling, yields only marginal improvements over the baseline model, whereas the ensemble approach achieves significantly better results. Although the individual performance of MofleNet57 and CPR59 models is notable, their true value lies in the significant savings on MACs and model size.

Development results demonstrate that the ensemble of the two models significantly improves accuracy by approximately 6% compared to the baseline.

6.2. Challenge Results

This section provides a critical analysis of the challenge results and submitted systems. Table 5 [14] displays the top team’s submissions and rankings, with our team (OO_NTUPRDCSG) securing 7th place. Notably, our ensemble of MofleNet with CP-ResNet demonstrates a robust strategy, yielding a 4% performance increase on the challenge test data compared to the baseline, while reducing the model size from 122 kB to 116kB. Unlike the top models, which employed Knowledge Distillation and external data, our model was trained directly on development dataset subsets, showcasing its effectiveness in handling limited data without relying on additional resources.

Analysis of the top-ranked models revealed that both the 1st and 2nd rank submissions were fine-tuned versions of CP-Mobile [4]. Replacing 3x3 convolutions with a combination of 1x3 and 3x1 convolutions reduced CP-Mobile model complexity but did not improve performance, indicating that model pruning was the key contributor to the top-ranked submission’s success. The 2nd place submission’s key contribution lies in its novel training strategy for CP-Mobile.

Our approach closely aligns with the 3rd-ranked submission, with the primary difference being their use of Knowledge Distillation. MofleNet and CP-ResNet ensemble achieved 64% accuracy, nearing the 3rd-ranked submissions on the development dataset with Knowledge Distillation. Our key contribution is the development and strong performance of MofleNet and its ensemble.

7. CONCLUSIONS

In this work, we presented our approach for Task 1: Low-Complexity Acoustic Scene Classification in the DCASE 2024 challenge. We introduced MofleNet, a novel hybrid architecture incorporating shuffle channels and residual inverted bottleneck blocks and used it in an ensemble with CP-ResNet. Our methods included augmentation techniques such as Freq-MixStyle and Device Impulse Response, along with Quantization Aware Training to meet the model size constraint. Our experimental results demonstrated that the ensemble of MofleNet and CP-ResNet significantly improved accuracy compared to individual models by approx. 4% and baseline by approx. 6%. Specifically, MofleNet performed better with larger datasets, while CPR59 was more effective with smaller datasets. Additionally, DCASE’24 Task 1 challenge results demonstrate the strength and potential of our ensemble approach. Despite not utilizing Knowledge Distillation, our model demonstrated good performance in handling limited data scenarios. This work highlights the importance of model ensemble and novel design of MofleNet, setting a foundation for future advancements in this domain.

8. REFERENCES

[1] F. Schmid, P. Primus, T. Heittola, A. Mesaros, I. Martín-Morató, K. Koutini, and G. Widmer, “Data-efficient low-complexity acoustic scene classification in the dcase 2024 challenge,” arXiv preprint arXiv:2405.10018, 2024.

[2] T. Heittola, A. Mesaros, and T. Virtanen, “Acoustic scene classification in dcase 2020 challenge: Generalization across devices and low complexity solutions,” in DCASE Workshop, 2020.

[3] F. Schmid, S. Masoudian, K. Koutini, and G. Widmer, “ Knowledge distillation from transformers for low complexity acoustic scene classification,” in DCASE Workshop, 2022.

[4] F. Schmid , T. Morocutti , S. Masoudian , K. Koutini , G. Widmer, “CP-JKU Submission to DCASE23: Efficient Acoustic Scene Classification with CP-Mobile,” in DCASE, 2023.

[5] T. S. Kim, D. Rho, G. Lee, and J. H. Park, “Dual-Strategy Enhancement of Acoustic Scene and Event Classification: Integrating Res2Net, GhostNet, and MobileFormer Architectures,” in DCASE, 2023

[6] Y. Cai, M. Lin, C. Zhu, S. Li, and X. Shao, “DCASE2023 Task 1 Submission: Device Simulation and Time-Frequency Separable Convolution for Acoustic Scene Classification,” in DCASE, 2023.

[7] J. Tan, and Y. Li, “Low-Complexity Acoustic Scene Classification Using Blueprint Separable Convolution and Knowledge Distillation,” in DCASE, 2023.

[8] B. Kim, S. Yang, J. Kim, H. Park, J. Lee, and S. Chang, “Domain Generalization with Relaxed Instance Frequency-wise Normalization for Multi-device Acoustic Scene Classification,” arXiv preprint arXiv:2206.12513, 2022.

[9] T. Morocutti, F. Schmid, K. Koutini, and G. Widmer, “Device-Robust Acoustic Scene Classification via Impulse Response Augmentation,” in 2023 31st European Signal Processing Conference (EUSIPCO), pp. 176-180. IEEE, 2023.

[10] “Microphone Impulse Response Project,” 2017. URL: https://micirp.blogspot.com/?m=1.

[11] X. Zhang, X. Zhou, M. Lin, J. Sun, “ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 6848-6856, 2018.

[12] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library, in Advances in Neural Information Processing Systems (NeurIPS),” advances in neural information processing systems 32, 2019.

[13] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, “Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2704-2713, 2018.

[14] “Data-Efficient Low-Complexity Acoustic Scene Classification 2024 Task 1 Challenge Results, 2024. URL: https://dcase.community/challenge2024/task-data-efficient-low-complexity-acoustic-scene-classification-results

[15] S. Woo, S. Debnath, R. Hu, X. Chen, Z. Liu, I. S. Kweon, and S. Xie, “Convnext V2: co-designing and scaling convnets with masked autoencoders,” CoRR, vol. abs/2301.00808, 2023.

IMPROVING LANGUAGE-BASED AUDIO RETRIEVAL USING LLM AUGMENTATIONS

Bartłomiej Zgórzyński, Jan Kulik, Juliusz Kruk, Mateusz Matuszewski

Samsung R&D Institute Poland, Warsaw, Poland
{b.zgorzynski, j.kulik, j.kruk, m.matuszews2}@samsung.com

ABSTRACT

This work explores the integration of large language models (LLMs) in multimodal machine learning, focusing on their usefulness in augmenting and generating audio-caption datasets. The study is structured around three primary objectives. The first objective is to evaluate the capability of LLMs to enhance existing audio-caption datasets by generating augmented and improved captions. The second objective explores the potential of LLMs to create new audio-caption datasets by extracting relevant text and audio from video-caption datasets. Various LLMs and hyperparameter configurations are tested to determine their effectiveness in these two tasks. The final objective is to evaluate the impact of these augmented and newly created datasets on training outcomes, providing insights into their potential contributions to audio related machine learning tasks. The results demonstrate the potential of LLMs to significantly advance the field by improving data quality and availability, in result enhancing model training and performance.

Index Terms— Language-Based Audio Retrieval, DCASE 2024, Large Language Models, Caption augmentation, Bi-encoder architecture, Multimodal learning

1. INTRODUCTION

The rapid advancement of multimodal machine learning has led to an increasing demand for high-quality and diverse audio-caption datasets. However, collecting such datasets by gathering audio samples and captioning them manually can be a time-consuming and resource-intensive task, often resulting in limited availability and quality issues. Recent developments in large language models have shown promising capabilities in natural language processing tasks, sparking interest in their potential applications in multimodal learning.

The effectiveness of utilizing text augmentations has been demonstrated by Primus et al. [1] through the application of various tools. However, with the advent of GPT-3 and subsequent advancements in large language models, it has become evident that these models alone can perform a range of complex augmentations, notably improving results of language-audio retrieval. The objective of this work is to evaluate and benchmark the effectiveness of augmentations performed by LLMs. We mainly focus on exploring two potential applications of large language models in language-audio retrieval task. The first is to augment captions using back-translation and mixing described in subsection 2.2. Primus et al. [2] showed that paraphrasing captions to Clotho v2.1 dataset using GPT-3.5 Turbo can successfully enrich the training data. Xu [3] introduced AudioSetMix, which employed LLM-assisted transformations for audio captions, demonstrating how LLMs can dynamically augment audio-caption datasets to improve both the diversity and quality of the data. In alignment with this, Wu et al. [4, 5] have proven that mixing captions improves the performance of audio captioning. Audio retrieval and captioning are in principle closely related [6] and therefore we further explore the effect of mixing captions on the former. The second LLM implementation is to generate a new audio-caption dataset from existing video-caption datasets by extracting audio descriptions from captions using LLMs as presented in subsection 2.3). The WavCaps [7] dataset utilizes LLMs to refine raw audio descriptions into more structured, caption-like sentences, demonstrating an effective use of LLMs in creating cleaner, more useful datasets for multimodal learning.

We further explore this concept by experimenting with different LLMs, hyperparameters and systematically measuring the results. The model we are using for comparison to verify and measure the results of proposed methods is a custom bi-encoder architecture inspired by the work of Primus et al. [2], and is described in subsection 2.1. In accordance with the objectives of this work, experiments are conducted exclusively using LLMs. Simpler methods of data augmentation are not tested, as LLMs are expected to yield more sophisticated and effective results. The conducted experiments and the final results are analyzed in section 3.

The findings presented in this work demonstrate promising results, warranting further investigation to fully explore the vast potential of LLMs in multimodal learning.

2. METHOD

2.1. Model training

To evaluate selected augmentations and data generation methods, we use a two-phase approach with an audio retrieval model: pre-training and fine-tuning. For pre-training, we employ datasets Clotho v2.1 [8], AudioCaps [9], and WavCaps [7]. We then fine-tune the model using only the Clotho and AudioCaps datasets. The effectiveness of the model is assessed by comparing the mean average precision at rank 10 (mAP@10) across test splits of these two datasets. The model consists of a bi-encoder architecture designed to estimate similarity between audio and text data. Input audio and text are mapped to a 1,024-dimensional latent space, where pairs with similar meanings are positioned close to each other, while pairs with different meanings are positioned further apart. The similarity between embeddings is determined using cosine similarity. For textual embeddings we utilize the RoBERTa-large model [10], and to encode audio we use the PaSST-S [11] encoder. We train the entire model simultaneously without freezing any layers.

To train our systems, we employ the InfoNCE loss with a trainable temperature. After calculating embeddings of all n audios and texts from a given batch, we compute the similarity matrix S, where S_{ij} denotes the similarity between text i and audio j. The diagonal of the matrix represents matching pairs, while all other elements are considered non-matching. We then calculate the mean

Original caption	Back-translated caption
Brakes squeak, and a quiet engine idles nicely	The brakes squeal, and a quiet engine slows down smoothly
Loud deep tone cascading through a large room	Deep and loud tone resonating in a large room
A siren wails into the open air while waves lap the shore	A siren sounds in the air while the waves hit the shore

Table 1: Examples of back-translation

cross-entropy loss on each row (text-to-audio loss) and each column (audio-to-text loss) after applying the softmax function. The final loss is the mean of the audio-to-text and text-to-audio components.

We analyze 30-second audio segments based on Clotho's maximal audio length. Since the audio encoder processes 10-second segments, we split the input audio into 10-second windows with a 10-second hop size, without any additional overlap between windows. Subsequently, we average all embeddings from a given audio to obtain final representation.

2.2. Augmentations

In this section, we explore two augmentation methods: back-translation and mixing. The back-translation method subtly modifies captions by translating them into a random language and then back to English, leveraging linguistic nuances to introduce minor yet impactful changes. Meanwhile, mixing involves combining audio samples and captions to generate new, coherent captions that expand our dataset significantly.

2.2.1. Back-translation

Each caption is translated to a random language and then back to English using a large language model. The following prompt was used for this task:

> *You will be given audio captions. The captions are going to be used for training of an audio captioning model. Translate every caption to a random language and then translate it back to English. When translating, feel free to make proper adjustments to ensure the phrase is natural and coherent. Do not comment on translations.*

At first glance, this method appears similar to simple paraphrasing; however, it offers two significant advantages. First, it introduces subtle yet noticeable modifications to the original text, leveraging the inherent differences between languages. Second, by operating within the constraints of translation, the LLM preserves the core meaning of the caption. Consequently, this approach produces slightly modified captions, often with a different word order, while minimizing the risk of altering the fundamental message. We present some examples of this augmentation on captions from Clotho v2.1 dataset using GPT-4o in Table 1.

2.2.2. Mixing

Audio samples from the Clotho and AudioCaps datasets are mixed with each other and the LLM is prompted to combine the corresponding captions in a sensible manner. This process results in the creation of 50,000 new audio-caption pairs. The following prompt was used as input for this task:

> *You will be given a list of audio captions. Your task is to mix them together to generate a new caption. The caption that you generate should be a mix of all the input captions. Keep the generated caption under 15 words. Do not write introductions or explanations. The caption should be a natural and coherent sentence in the style of the input captions. The captions are not chronological, so don't refer to time dependencies between them.*

2.3. VideoCaps

2.3.1. Dataset generation

In order to create a new high-quality dataset, we collected commonly used video-caption datasets: Activity-Net [12], Charades-Ego [13], MSRVTT [14], MSVD [15], VATEX [16], VIOLIN [17] and WebVid [18]. This resulted in obtaining around 10.8 million samples. Then, we extracted samples that contained valid audio, which narrowed the dataset down to around 770,000 audio-caption pairs. The main challenge was that many of the captions were primarily video-focused and did not contain any meaningful information about the audio content.

Therefore, in order to filter out such cases, we employ the fine-tuned model described in subsection 2.1 to obtain audio and text embeddings for each sample. Then, cosine similarity between embeddings of each ground-truth pair is computed. This approach leverages the model's capability to represent complex semantic relationships and can be used to estimate the quality of ground-truth pairs in an arbitrary dataset. This method was applied to select top 100,000 samples for further processing.

To further process the selected samples, the LLM was used to rephrase original captions and remove any visual context that would be irrelevant during audio retrieval training. The following prompt was used as input to the LLM:

> *You will be given video captions. Rephrase them and remove parts that couldn't possibly be inferred from audio events. Remove any details from the captions that refer to visual or spoken events. Focus on the audio content only. Remove dates, time and names of places and persons. Do not write introductions or explanations. Each audio caption should be one sentence with less than 15 words. Use grammatical sentences.*

Finally, since rephrasing is prone to outliers and low-quality results as the LLM may deem the input captions as inadequate or simply fail to perform the task properly, we perform final filtration on the rephrased captions and extract the top 70,000.

2.3.2. Selecting LLM temperature

The aforementioned method of evaluating dataset quality can also serve as a benchmark to evaluate performance of various LLMs in processing captions and aid in selecting hyperparameters. The latter is especially important, since temperature can have significant impact on the quality of LLM output [19] and its optimal value can only be determined empirically. Therefore, we conducted a grid search and used various commercial and open-source LLMs with different temperature settings to rephrase 1,000 captions that were randomly sampled from the top 100,000 pairs obtained earlier. Then, cosine similarity between each rephrased caption and the corresponding audio clip was computed.

Experiment	LLM used	AudioCaps mAP@10	Clotho mAP@10
Pre-training	-	56.70	37.36
Pre-training + VideoCaps	GPT-4o	56.73	38.12
Pre-training + VideoCaps (without WavCaps)	GPT-4o	54.77	34.21
Base fine-tuning	-	59.43	38.68
Back-translation	Llama 3 8B	59.10	38.95
Back-translation	GPT-3.5 Turbo	59.76	39.14
Back-translation	GPT-4o	59.71	39.11
Mixing	Llama 3 8B	**60.61**	39.17
Mixing	GPT-3.5 Turbo	59.23	39.18
Mixing	GPT-4o	59.81	**39.24**
VideoCaps	Llama 3 8B	58.57	38.70
VideoCaps	GPT-3.5 Turbo	58.57	38.56
VideoCaps	GPT-4o	58.87	38.44
VideoCaps + Mixing	GPT-4o	59.08	38.82
VideoCaps + Back-translation	GPT-4o	59.38	39.02
Back-translation + Mixing	GPT-4o	59.44	38.94
VideoCaps + Mixing + Back-translation	GPT-4o	58.87	38.44

Table 2: Performance of text-to-audio retrieval on the AudioCaps and Clotho v2.1 test sets was evaluated. Each model was trained three times, with the values reported in the tables representing the average performance on each dataset.

3. EXPERIMENTS AND RESULTS

In this section, we describe all conducted experiments. Table 2 presents the performance of all models on the Clotho v2.1 and AudioCaps datasets, including both pre-training and fine-tuning results. For all experiments, we utilize the InfoNCE loss function. We update the model parameters using the AdamW optimizer with a batch size of 128. Additionally, we employ a cosine decay learning rate scheduler with warmup. During training, we select the best model checkpoints based on the mAP@10 value evaluated on the validation set, which is assessed twice within each training epoch.

3.1. Pre-training

Initially, we aimed to develop an audio-retrieval model for subsequent fine-tunings and data filtering. The training phase utilized Clotho-training, AudioCaps-training, and WavCaps datasets, with Clotho-validation and AudioCaps-validation employed for validation purposes. The training consists of 16 epochs, with a learning rate schedule from 1×10^{-5} to 5×10^{-7}. We utilize structured patchout of 15 and 2 for time and frequency dimensions, respectively. Additionally, random deletion and synonym replacement are applied with a probability of 0.8.

3.2. Fine-tuning

The next step involves further fine-tuning the model. For this purpose, the same datasets were used for both training and validation, excluding WavCaps. The number of epochs has been reduced to 6, and the learning rate has been decreased to range from 3×10^{-6} to 6×10^{-8}. To increase model regularization, we changed the optimizer weight decay parameter from 0.0 to 0.1. The results indicate that additional fine-tuning without WavCaps significantly increases the mAP@10 on both the Clotho and AudioCaps datasets.

3.3. Back-translation and mixing

A certain degree of randomness in generation is particularly desirable, especially during back-translation, to avoid literal translation. We decided to set the temperature parameter to 0.7 for each of the LLMs. Then, we have prepared augmented datasets: for each caption in training splits of AudioCaps and Clotho v2.1 we have generated exacly one back-translated caption. For mixing, we randomly selected 50,000 data pairs, equalized their audio energies, and used the LLMs to combine their captions.

To evaluate the effectiveness of augmented datasets in the development of audio retrieval systems, we conducted additional fine-tuning experiments. The effectiveness of these augmentations was assessed by comparing the mAP@10 value on the AudioCaps and Clotho v2.1 datasets, based on the training data used. The results, presented in Table 2, demonstrate that both back-translation and mixing significantly enhance the model's performance. For back-translation, utilizing more advanced language models leads to improved results, while for mixing, the best results were obtained with the smallest tested model.

Table 3 demonstrates that our best model, enhanced through data augmentation using Llama 3 8B, outperforms most of current state-of-the-art solutions. The single models by Primus et al. and Chen et al., submitted to the DCASE 2023 and 2024 Challenges, were trained on the full AudioCaps dataset and validated exclusively on the Clotho v2.1, which naturally resulted in improved performance on this dataset.

3.4. VideoCaps

3.4.1. Temperature selection

To select the optimal temperature settings for each large language model, we conducted experiments across a spectrum of temperature settings, ranging from 0.0 to 1.5. These settings adjust the randomness in the model's output: lower temperatures result in more deterministic outputs, while higher temperatures allow for greater

Model	AudioCaps mAP@10	Clotho mAP@10
CLAP [20]	51.3	20.4
Chen et al. [21]	-	37.00
Primus et al. [2]	-	38.56
Primus et al. [22]	-	**39.77**
Ours	**60.61**	39.17

Table 3: Comparison of our solution with other state-of-the-art text-to-audio retrieval systems.

diversity but potentially less coherence and relevance to the original audio content. We measured the median cosine similarity of 1,000 selected samples after rephrasing.

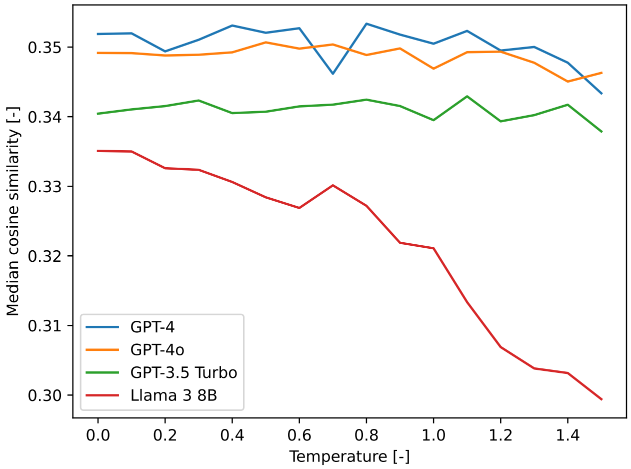


Figure 1: Median cosine similarity of rephrased datasets obtained using various LLMs across a spectrum of temperature settings

The results, shown in Figure 1, indicate that all models demonstrate higher median cosine similarity at lower temperatures. This finding suggests that more deterministic settings produce captions that are more closely aligned with the reference captions, highlighting the trade-off between creativity and accuracy in model-generated captions. Additionally, GPT-4 and GPT-4o consistently outperform GPT-3.5 Turbo and Llama 3 8B across most temperature settings, suggesting that newer and more sophisticated model architectures may better maintain semantic accuracy even as the output becomes more diverse at higher temperatures. For further experiments, we selected a temperature of 0.0 for Llama 3 8B, 1.1 for GPT-3.5 Turbo, and 0.7 for GPT-4o.

3.4.2. Trainings

After filtering, VideoCaps contains approximately 70,000 new audio-caption pairs. To evaluate the dataset's quality, we conducted additional pre-training to measure the influence of VideoCaps on model performance, keeping pre-training settings the same. The results of this experiment are shown in Table 2. Substituting WavCaps with VideoCaps alone resulted in a significant performance decrease on both datasets. However, integrating both WavCaps and VideoCaps notably enhanced performance on Clotho v2.1 while maintaining comparable results on AudioCaps.

We also conducted fine-tuning after base pre-training. The results showed slightly lower performance compared to fine-tuning without VideoCaps. This difference may be attributed to the larger volume of VideoCaps data compared to AudioCaps and Clotho v2.1. Additionally, variations in performance could stem from the fine-tuning process adapting to specific styles of descriptions and audio content present in the evaluation sets.

3.5. Joint Trainings

We also conducted experiments using different augmentations during fine-tuning, exclusively utilizing data generated by GPT-4o. The results are shown in Table 2, revealing that none of the combinations surpassed the performance achieved with Mixing and Llama 3 8B.

4. DISCUSSION AND CONCLUSION

In this study we explored various approaches to augmenting and generating new datasets for the training of text-to-audio retrieval systems using large language models. The results demonstrate that the utilization of presented techniques can significantly enhance retrieval model performance.

The novel approach introduced in this paper by creating VideoCaps dataset, shows promising results for generating large-scale text-audio datasets, thereby improving model pre-training. We also showed that large language model choice and the value of the temperature parameter can significantly impact the quality of the generated dataset. Additionally, our experiments indicate that mixing audio and captions, especially when augmented using Llama 3 8B, yields the best results for our system. This method produced a new state-of-the-art model in text-to-audio retrieval, achieving a mAP@10 score of 60.61 on the AudioCaps dataset. Additionally, it achieved a comparable performance to the current state-of-the-art models on the Clotho v2.1 dataset, with a score of 39.17.

However, fine-tunings with multiple generated data resulted in lower performance. One possible explanation for this is that there is a larger quantity of artificially generated data compared to the original data, which are likely of better quality.

5. FUTURE RESEARCH

In our research, we introduced three distinct ways to create and enhance datasets using large language models. There is still a room for experiments and improvements.

In addition to our methods, there are alternative approaches to generating synthetic audio-caption datasets. One such approach involves using the outputs of an audio captioning model as part of the prompt, along with other relevant metadata. Another method is to mix audio clips sequentially without significant overlap and prompt the LLM to generate corresponding captions, taking into account the temporal aspects of the audio.

Further research should investigate the outcomes of utilizing different LLMs and fine-tuning their hyperparameters, such as temperature, top-p, and top-k. In addition, experimenting with prompt engineering seems be an essential approach. It is reasonable to assume that as the quality of available large language models improves over time, the quality of synthetically generated datasets will also enhance.

6. REFERENCES

[1] P. Primus and G. Widmer, “Improving natural-language-based audio retrieval with transfer learning and audio & text augmentations,” 2022. [Online]. Available: https://arxiv.org/abs/2208.11460

[2] P. Primus, K. Koutini, and G. Widmer, “Advancing natural-language based audio retrieval with passt and large audio-caption data sets,” in *Proceedings of the 8th Detection and Classification of Acoustic Scenes and Events 2023 Workshop (DCASE2023)*, Tampere, Finland, September 2023, pp. 151–155.

[3] D. Xu, “Audiosetmix: Enhancing audio-language datasets with llm-assisted augmentations,” 2024. [Online]. Available: https://arxiv.org/abs/2405.11093

[4] S.-L. Wu, X. Chang, G. Wichern, J.-w. Jung, F. Germain, J. L. Roux, and S. Watanabe, “Beats-based audio captioning model with instructor embedding supervision and chatgpt mix-up,” DCASE2023 Challenge, Tech. Rep., May 2023.

[5] S.-L. Wu, X. Chang, G. Wichern, J. weon Jung, F. Germain, J. L. Roux, and S. Watanabe, “Improving audio captioning models with fine-grained audio features, text embedding supervision, and llm mix-up augmentation,” 2024. [Online]. Available: https://arxiv.org/abs/2309.17352

[6] E. Labbé, T. Pellegrini, and J. Pinquier, “Killing two birds with one stone: Can an audio captioning system also be used for audio-text retrieval?” in *Proceedings of the 8th Detection and Classification of Acoustic Scenes and Events 2023 Workshop (DCASE2023)*, Tampere, Finland, September 2023, pp. 86–90.

[7] X. Mei, C. Meng, H. Liu, Q. Kong, T. Ko, C. Zhao, M. D. Plumbley, Y. Zou, and W. Wang, “Wavcaps: A chatgpt-assisted weakly-labelled audio captioning dataset for audio-language multimodal research,” 2023.

[8] K. Drossos, S. Lipping, and T. Virtanen, “Clotho: An audio captioning dataset,” 2019.

[9] C. D. Kim, B. Kim, H. Lee, and G. Kim, “Audiocaps: Generating captions for audios in the wild,” in *NAACL-HLT*, 2019.

[10] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “Roberta: A robustly optimized BERT pretraining approach,” *CoRR*, vol. abs/1907.11692, 2019. [Online]. Available: http://arxiv.org/abs/1907.11692

[11] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, “Efficient training of audio transformers with patchout,” in *Interspeech 2022*. ISCA, 2022. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2022-227

[12] B. G. Fabian Caba Heilbron, Victor Escorcia and J. C. Niebles, “Activitynet: A large-scale video benchmark for human activity understanding,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 961–970.

[13] G. A. Sigurdsson, A. Gupta, C. Schmid, A. Farhadi, and K. Alahari, “Charades-ego: A large-scale dataset of paired third and first person videos,” 2018.

[14] G. Tan, D. Liu, M. Wang, and Z.-J. Zha, “Learning to discretely compose reasoning module networks for video captioning,” 2020.

[15] D. L. Chen and W. B. Dolan, “Collecting highly parallel data for paraphrase evaluation,” in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL-2011)*, Portland, OR, 6 2011.

[16] X. Wang, J. Wu, J. Chen, L. Li, Y.-F. Wang, and W. Y. Wang, “Vatex: A large-scale, high-quality multilingual dataset for video-and-language research,” 2020.

[17] J. Liu, W. Chen, Y. Cheng, Z. Gan, L. Yu, Y. Yang, and J. Liu, “Violin: A large-scale dataset for video-and-language inference,” 2020.

[18] M. Bain, A. Nagrani, G. Varol, and A. Zisserman, “Frozen in time: A joint video and image encoder for end-to-end retrieval,” in *IEEE International Conference on Computer Vision*, 2021.

[19] M. Peeperkorn, T. Kouwenhoven, D. Brown, and A. Jordanous, “Is temperature the creativity parameter of large language models?” 2024.

[20] Y. Wu, K. Chen, T. Zhang, Y. Hui, M. Nezhurina, T. Berg-Kirkpatrick, and S. Dubnov, “Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation,” 2024. [Online]. Available: https://arxiv.org/abs/2211.06687

[21] M. Chen, Y. Liu, B. Peng, and J. Chen, “Dcase 2024 challenge task 8 technical report,” DCASE2024 Challenge, Tech. Rep., June 2024.

[22] P. Primus and G. Widmer, “A knowledge distillation approach to improving language-based audio retrieval models,” DCASE2024 Challenge, Tech. Rep., June 2024.