CONTINUAL LEARNING FOR MUSIC CLASSIFICATION

Pedro González-Barrachina[1,2] **María Alfaro-Contreras**[1] **Jorge Calvo-Zaragoza**[1]

[1] Pattern Recognition and Artificial Intelligence Group, University of Alicante, Spain
[2] Alice Biometrics

pcg71@alu.ua.es,{malfaro, jcalvo}@dlsi.ua.es

ABSTRACT

Music classification is a prominent research area within Music Information Retrieval. While Deep Learning methods can adequately perform this task, their classification space remains fixed once trained, which conflicts with the dynamic nature of the ever-evolving music landscape. This work explores, for the first time, the application of Continual Learning (CL) in the context of music classification. Specifically, we thoroughly evaluate five state-of-the-art CL approaches across four different music classification tasks. Additionally, we showcase that a foundation model might be the key to CL in music classification. To that end, we study a new approach called *Pre-trained Class Centers*, which leverages pre-trained features to create fixed class-center spaces. Our results reveal that existing CL methods struggle when applied to music classification tasks, whereas this simple method consistently outperforms them. This highlights the need for CL methods tailored specifically for music classification.

1. INTRODUCTION

Music Information Retrieval (MIR) is a multidisciplinary field dedicated to retrieving information from music sources [1]. Within the MIR domain, music classification stands as one of the most widespread research topics [2]. It involves the categorization of music into various predefined classes, with these categories defining the ultimate task at hand. There is a diverse range of classification tasks, including genre classification [3], vocal technique identification [4], instrument classification [5], and singer identification [4], among others. These tasks are essential for organizing and retrieving music efficiently, enabling applications such as recommender systems and music search engines to better serve the needs of users in the ever-evolving music landscape [6].

Traditional music classification approaches predominantly relied on signal processing methods, accompanied by heuristics and handcrafted features, to categorize music data [7, 8]. However, these schemes often struggled to capture the complex and nuanced aspects of musical content, thus limiting their practical application. With the rise of Deep Learning (DL) strategies, alternative solutions emerged to ease this task [9, 10]. DL models address these issues by automatically learning hierarchical representations from the data itself, thereby improving the accuracy and flexibility of music classification systems.

However, DL models become static once they are trained; their feature space is fixed. Consequently, they may struggle or fail to accommodate new classes. This does not align well with the dynamic nature of music itself—characterized by evolving genres, emerging artists, and shifting musical trends. We could approach this challenge in two ways: either (i) retrain the model from scratch when new music data is introduced, which is computationally expensive, inefficient, and not always possible due to privacy or storage issues [11], or (ii) fine-tune the model only on the newly acquired music data. The latter alternative is known to lead to the so-called "catastrophic forgetting", where the knowledge acquired from previous data diminishes as new information is incorporated [12]. This situation highlights the need for robust and adaptable music classification systems that can be updated with just new data.

Continual Learning (CL) promises a solution to catastrophic forgetting by enabling models to gradually incorporate new knowledge without forgetting previously acquired information [11, 13]. Fig. 1 graphically depicts this scenario. This adaptability is vital for music classifiers to stay up-to-date, ensuring they can accurately categorize a continuously evolving musical landscape. While some previous works in zero-shot [14] and few-shot learning [15] propose methods for recognizing new, unseen classes, they do not maintain nor update the knowledge acquired in one session in subsequent sessions. In contrast, our work introduces the use of CL in music classification, with the goal of not only recognizing unseen classes but also retaining this knowledge over time.

CL approaches are generally classified according to the following taxonomy [16]: (i) *data-centric* methods, which focus on preserving important data from previous tasks using *data replay* or *data regularization* techniques; (ii) *model-centric* methods, which focus on model development through *parameter regularization* or model structure

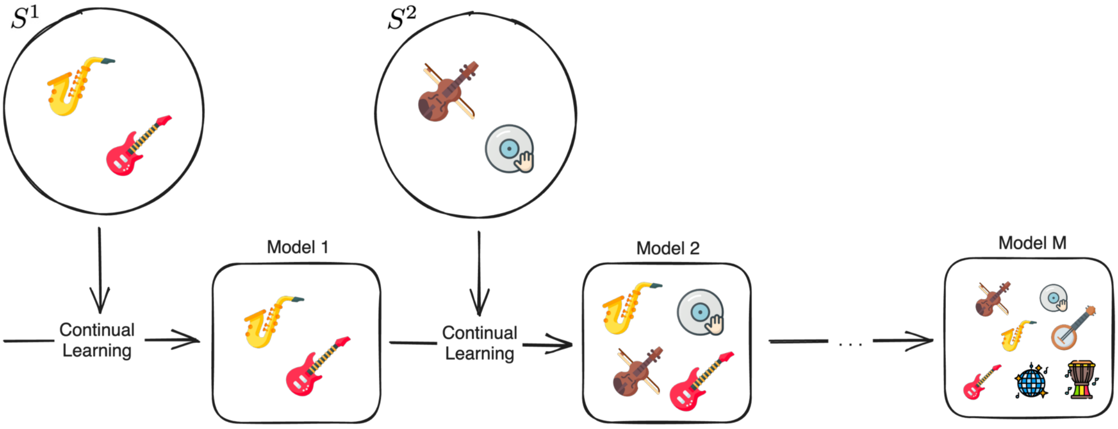


Figure 1. Graphical representation of class-incremental learning for music classification tasks. The process begins with Model 1, trained to differentiate some initial classes during session S^1. Through a continual learning algorithm, Model 1 preserves its acquired knowledge while incorporating new classes in a subsequent learning session, S^2, thus evolving into Model 2. This iterative learning process continues, enabling the model to progressively expand its repertoire of recognizable classes.

expansion, i.e., *dynamic networks*; and, (iii) *algorithm-centric* methods, which focus on the learning process itself, employing *knowledge distillation* techniques or rectifying *model biases*.

In this work, we investigate the applicability of state-of-the-art CL techniques, originally designed for computer vision, in the context of music classification tasks. Additionally, we explore the use of foundation models to introduce a new CL method based on pre-trained representations to create fixed class-centers, showcasing the utility and robustness of foundational models. Our results reveal that existing CL methods, traditionally evaluated on image classification, struggle when applied to music, whereas our proposed method consistently outperforms them. This raises questions regarding the effectiveness and transferability of existing CL techniques to music classification. Moreover, it prompts us to consider whether leveraging foundational models might represent a better approach for addressing the CL paradigm in certain scenarios.

To summarize, the contributions of this work are as follows: (i) a first-time analysis of the applicability of CL techniques to music classification tasks, (ii) the introduction of a simple yet effective CL method that relies on the generalizability of large pre-trained models, and (iii) extensive experimentation to quantitatively evaluate five different CL approaches across four music classification benchmarks with two different pre-trained feature extractors.

2. METHODOLOGY

In this work, we address the music classification task from a CL perspective, with a specific focus on Class-Incremental Learning (CIL). In each learning *session*, the model is trained with new audio tracks from a new set of classes. Ideally, the model should learn to classify the new classes introduced in each session while retaining its capacity to classify classes from previous sessions (see Fig. 1).

Formally, let us assume a sequence of M training sessions $\{S^1, S^2, \cdots, S^M\}$, where each session has a different set of non-overlapping classes. $S^m = (X_m, Y_m)$ represents the m-th incremental step, with X_m containing audio tracks whose labels belong to Y_m, and Y_m denoting the label space of session m, where $Y_m \cap Y_{m'} = \oslash$ for $m \neq m'$. Note that the audios in X_m are in the format $[0, 1]^{l_j \times c}$, being l_j the length of the j-th audio.[1] In this work, we consider mono audio signals as input ($c = 1$), although other considerations may be applicable. After each session, the model is evaluated on all seen classes $\Upsilon_m = Y_1 \cup \cdots Y_m$. The main objective of CIL is to sequentially build a classification model capable of classifying all seen classes.

2.1 Classification model

For this learning framework, our classification model consists of a fixed pre-trained model, serving as the feature extractor, and an out-of-the-box fully connected network, acting as the downstream task classifier. We use this *same* learning framework with different CL strategies to compare their performance. In order to make our experimentation agnostic to a certain degree to the pre-trained model selected as the feature extractor, we consider two state-of-the-art pre-trained models:

1. **MERT** [10] is a recently released foundational model specifically designed for extracting rich representations from music data. It follows a self-supervised pre-training paradigm that relies on two teacher models, one for the acoustic aspect and one for the musical aspect, to generate pseudo-labels for sequential audio clips. This multi-task paradigm allows for a balanced acoustic and musical representation learning, guiding a BERT-style transformer encoder to better model music audio. Its state-of-the-art performance across various MIR tasks, including

[1] Audio chunks of l_j are considered to accommodate different lengths, as typically done in the literature.

those relevant to this work, makes it a compelling choice for our purposes.

2. **CLMR** [17] adapts the image self-supervised learning strategy SimCLR [18] to the domain of music. This method employs contrastive learning to train a convolutional feature extractor [19] to extract meaningful and transferable representations from music data. It achieves this by learning to predict similar representations for slightly altered versions of the same audio sample. We consider CLMR to be a robust classification model for our work, given its demonstrated effectiveness across various music classification tasks, along with its lightweight architecture.

A summary of the characteristics of these two models can be seen in Table 1. Both of these pre-trained models use raw audio samples as inputs. We chose them with the presumption that their differences in architecture and size would enable us to extract more nuanced insights and conclusions from our experiments. It is worth noting that, in order to have comparable results with recent research works, we adhere to the same evaluation protocol as outlined in [10].

Table 1. Overview of the feature extractor models used in this work, depicting their characteristics (architecture, number of trainable parameters, input audio length, and feature embedding size).

	Architecture	Audio length (s)	Embedding Size
MERT	Transformer (94.9M parameters)	5	764
CLMR	CNN (2.4M parameters)	2.7	512

2.2 Selected methods

We select a diverse range of state-of-the-art methods in CL, emphasizing the inclusion of methods from different subtypes across the entire taxonomy. Specifically, we consider five CL approaches:

1. **Replay** aims to prevent catastrophic forgetting by employing a data-centric approach, which involves revisiting past data during the learning process [20].
2. **GEM** adopts a data-centric strategy based on data regularization to stabilize continuous training. It constrains the model's parameter updates to prevent significant forgetting of previously learned tasks, ensuring a balanced learning experience over time.
3. **EWC** employs a model-centric approach through parameter regularization [21]. It assigns importance to specific parameters based on their relevance in previously learned tasks, thereby preventing excessive adjustments during subsequent training on new tasks.
4. **L2P** utilizes a model-centric approach based on dynamic networks [11]. It aims to learn to prompt a pre-trained Transformer to adapt it to the new tasks, managing both task-invariant and task-specific knowledge while maintaining model plasticity. [2]
5. **iCaRL** adopts an algorithm-centric strategy of knowledge distillation [22]. It leverages distillation from frozen models of past learning sessions, combined with data replay, to avoid forgetting.

As a baseline method, we fine-tune the model for each session without applying a CL strategy, referred to as **Finetune** in the experiments, following the fine-tuning protocol used in state-of-the-art research [11]. This serves as our lower bound, potentially leading to the strongest occurrence of catastrophic forgetting.

2.3 Pre-trained Class Centers

In addition to the CL methods considered, we explore a novel approach that relies on the generalizability of the representations of a foundation model. We use this method to showcase the potential efficacy of using pre-trained models with self-supervised learning for CL. The idea is to use the latent representations produced by pre-trained models to capture the underlying semantics of the data itself, causing these representations to be distributed in a way that enables classification. Our approach seems particularly well-suited for music classification tasks because there exist publicly available foundation models (e.g., MERT and CLMR) known for their strong generalization capabilities. The proposed method, termed as *Pre-trained Class Centers* (PCC), can be separated into three different stages:

1. **Feature Extraction**. We extract a pre-trained feature embedding for each training sample.
2. **Prototype Generation**. We compute *prototype* class-centers by averaging all the feature embeddings obtained for each class in a training session and store them in a prototype buffer.
3. **Similarity Calculation**. During the inference phase, the class of a given test audio track is determined by the class associated with the nearest class-center, calculated through the Euclidean distance between the pre-trained feature vector of the test audio and the class-center prototype.

Although PCC is conceptually simple, it has never been considered. The method belongs to the data-centric category of CL methods because it focuses on leveraging the generalizability of pre-trained representations. One notable advantage is its memory efficiency (only one prototype per class), making it suitable for scenarios with limited computational resources. Just as important, this method stores representations rather than the original data, thus avoiding privacy issues. Furthermore, in PCC, the training process for each new class is independent of the other classes, making it a robust method for CL.

[2] Given that this method assumes a Transformer architecture as the backbone, it cannot be evaluated with CLMR.

3. EXPERIMENTATION

This section encompasses the experimental setup, including the music classification tasks, evaluation protocol, and implementation details.

3.1 Tasks

To conduct an extensive and diverse analysis, we evaluate the selected CL methods on four distinct music classification tasks using three different datasets:

Genre classification estimates the most appropriate genre for a given song. We use the standard curated split of the GTZAN dataset [23, 24], which consists of 930 30-second audio tracks from 10 different genres.

Instrument classification determines the specific musical instrument present within a given sound. We consider the NSynth dataset [5], which contains 306 000 4-second audio samples of an instrument playing a single note. There are 11 instrument classes in this dataset. Due to the high computational cost associated with the large size of the training partition,[3] we consider only 5 000 training samples for each class while keeping the validation and test sets intact.

Singer identification classifies the identity of a given vocal performer in an audio track. We employ the VocalSet dataset [4], which comprises 3 613 recordings of variable length from 20 professional singers performing using different vocal techniques.

Vocal technique detection recognizes the specific singing technique present within a given audio recording. We resort to the aforementioned VocalSet dataset, considering a subset of 10 different singing techniques, consisting of 1 736 audio samples, similar to referenced work [4].

For all tasks, we consider a sequence of $M = 5$ training sessions. Given a task comprised of C different classes,[4] each session will have an equally distributed randomly selected subset of C/M non-overlapping new classes. To avoid any bias related to the order of the sessions or the order in which the classes are learned, we report the average performance over three scenarios, each with a different sequence of training sessions. In each scenario, we randomly arrange the classes and create random groups of C/M classes. Our goal is to obtain a better estimate of the expected performance of the CL methods under unknown learning situations. Table 2 provides a summary of the characteristics of CL paradigm posed for each task.

3.2 Implementation details

As mentioned in Section 2, our classification model comprises two fundamental components: a feature extractor, which can be either a MERT or CLMR model, and a downstream task classifier. The feature extractors are used out-of-the-box.[5] [6] These remain frozen during training not only for efficiency but also to improve stability and mitigate the effects of forgetting in CL. The classifier is an MLP with 512 hidden units. When using MERT, a one-dimensional convolutional layer is employed prior to the MLP to extract a weighted average embedding from the frame-level features obtained by MERT.

Table 2. Overview of the continual learning scenario posed for each music classification task: the number of learning sessions, the total number of classes, and the number of classes per session.

Classification task	**Number of learning sessions**, M	**Total number of classes**, C	**Classes per learning session**, C/M
Genre	5	10	2
Instrument		11	2*
Singer		20	4
Vocal Technique		10	2

*The remaining class is randomly introduced in one of the learning sessions, i.e., there is one session with 3 classes.

We follow the details provided in the work of Li et al. [10] to train the architecture described previously for the different considered tasks. To attain state-of-the-art results while keeping the feature extractor frozen, we train the downstream classifier for a maximum of 200 epochs using the ADAM optimizer with a fixed learning rate of 10^{-3} and a batch size of 64 audio chunks. We use early-stopping with the number of patience epochs adjusted accordingly to each task. Additionally, we employ a 25% dropout rate to mitigate overfitting and improve performance.

For the methods that require data storage from past sessions (Replay, GEM, iCaRL), we use a memory buffer of 100 memories equally distributed among the classes seen up to that session, following the implementation used in [25]. Moreover, we use PyTorch as the implementation framework. We rely on the PyCIL toolbox[7] for all the considered CL methods, except for L2P, for which we adhere to the official implementation.[8]

The length of the audio chunks used for training and evaluating the models depends on the feature extractor used and can be seen in Table 1. For the task of singer identification and vocal technique, we use 3-second audio chunks as input, as in previous works [4, 10]. Finally, regarding the evaluation protocol, we segment each audio file into chunks (as aforementioned) and obtain a prediction for each chunk. The predictions for each chunk are then averaged to obtain a final prediction for each given audio file.

4. RESULTS

Fig. 2 reports the average performance of each method for each learning session in terms of classification accuracy.[9]

[3] For each task, we launched 212 training processes following the experimental setup considered (2 feature extractors × (3 scenarios × 7 CL methods × 5 sessions + 1 oracle baseline)).

[4] Each dataset is balanced, i.e., the same number of samples, or a very similar number, is considered for each class.

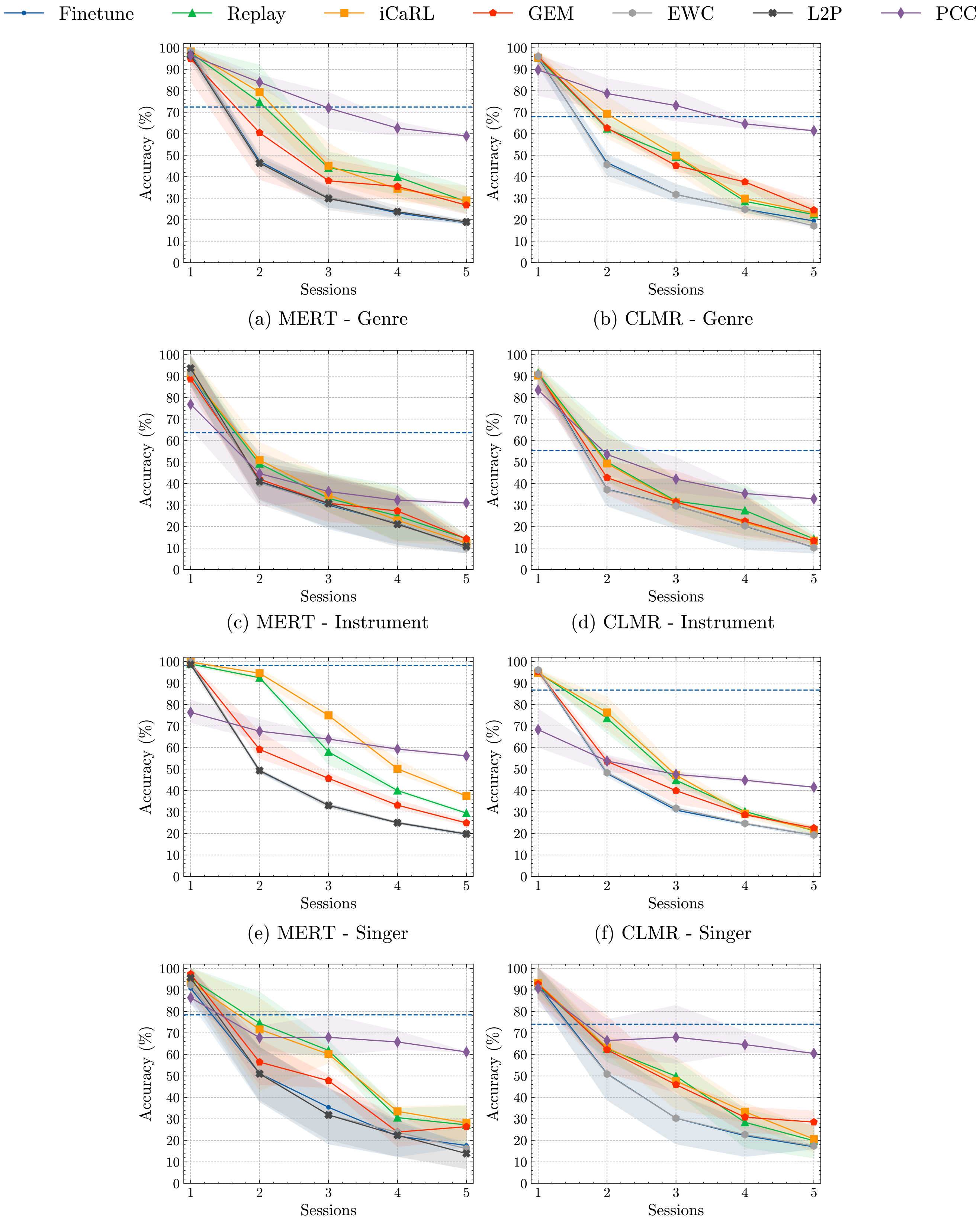
[5] MERT's weights available at https://huggingface.co/m-a-p/MERT-v1-95M

[6] CLMR's weights available at https://github.com/Spijkervet/CLMR

[7] https://github.com/G-U-N/PyCIL

[8] https://github.com/google-research/l2p

[9] The code developed in the work is publicly available for reproducible research at: https://github.com/pedrocg42/continual-music-classification



(a) MERT - Genre

(b) CLMR - Genre

(c) MERT - Instrument

(d) CLMR - Instrument

(e) MERT - Singer

(f) CLMR - Singer

(g) MERT - Vocal Technique

(h) CLMR - Vocal Technique

Figure 2. Accuracy (%) per session for each CL method. The solid lines represent the average accuracies, while the shaded areas indicate the minimum and maximum accuracies for each method and session. The dashed line represents the reference accuracy achieved when directly training with all classes in a single session.

We report as well the average accuracy along all the training sessions in Table 3 and the accuracy after the last session in Table 4.

Table 3. Comparison of the averaged accuracy after each session across the four tasks.

Task	**Genre**		**Instrument**		**Singer**		**Vocal Tech**	
Encoder	MERT	CLMR	MERT	CLMR	MERT	CLMR	MERT	CLMR
Finetune	43.4	43.6	38.6	37.9	45.3	43.7	43.4	42.6
Replay	56.9	51.6	42.4	43.0	63.8	53.0	57.9	50.6
iCaRL	57.2	53.5	42.4	41.3	**71.4**	**53.8**	57.4	51.6
GEM	51.1	53.2	40.6	40.2	52.4	48.0	50.4	52.0
EWC	43.5	43.1	38.7	37.6	45.4	44.0	43.1	42.4
L2P	43.1	-	39.5	-	45.1	-	42.9	-
PCC	**74.8**	**73.5**	**44.2**	**49.5**	64.6	51.2	**69.8**	**70.0**
Oracle	75.2	70.9	62.7	57.1	97.8	86.0	76.3	72.9

Table 4. Comparison of the final accuracy after the last session across the four tasks.

Task	**Genre**		**Instrument**		**Singer**		**Vocal Tech**	
Encoder	MERT	CLMR	MERT	CLMR	MERT	CLMR	MERT	CLMR
Finetune	18.5	19.4	10.1	10.3	19.7	19.3	17.6	17.0
Replay	28.4	22.4	14.4	14.3	29.5	21.3	27.2	19.8
iCaRL	29.0	23.1	12.4	13.4	37.5	21.6	28.1	20.6
GEM	26.8	24.5	14.3	13.4	24.9	22.5	26.4	28.5
EWC	18.6	17.1	10.1	10.1	19.8	19.3	16.2	17.4
L2P	19.0	-	10.9	-	19.8	-	13.9	-
PCC	**59.0**	**61.4**	**31.0**	**32.9**	**56.1**	**41.5**	**61.1**	**60.5**
Oracle	72.4	67.9	63.7	55.4	98.2	86.7	78.4	74.0

The first observation is the limitation of the existing CL literature, where methods are primarily evaluated over well-established computer vision tasks [11]. As evidenced by the reported results, such methods fall short in terms of generalization to other domains. Specifically, the considered state-of-the-art CL methods suffer from significant catastrophic forgetting, resulting in poor final performance for class-incremental music classification. This underscores the need to develop CL methods for this specific domain and encourages the assessment of CL methods across different fields to measure their overall performance more precisely.

Focusing our attention on the similarities illustrated in Fig. 2 for the two different feature extractors, MERT (left column) and CLMR (right column), we can observe that the accuracy curves for all tasks exhibit very similar trends across sessions. While the baseline performance—training directly with all data in a single session—is better when using MERT, this difference between the two feature extractors diminishes when comparing against the different CL methods, as similar performance is achieved. Consequently, we can conclude that the effectiveness of the CL methods is not solely attributable to the feature extractor.

If we examine each task separately, we observe a similar pattern in both music genre and vocal technique classification tasks. State-of-the-art methods exhibit signs of catastrophic forgetting, whereas PCC achieves a final performance that is relatively close to the reference bound. For singer identification, PCC starts with a lower accuracy but maintains good stability throughout the sessions. However, despite achieving the highest final performance among the methods, it still falls considerably short of the task reference. Finally, instrument classification emerges as the most challenging task, with all methods displaying significant signs of catastrophic forgetting. As a result, their final performance remains far from reaching reasonable results, once again highlighting the existing room for improvement and the need to find new methods that can reduce catastrophic forgetting in music classification.

Among the considered state-of-the-art CL methods, both data-centric (Replay and GEM) and algorithm-centric (iCaRL) approaches outperform the results obtained by model-centric methods (EWC and L2P). However, it is worth noting that these first three methods rely on input data stored from previous sessions, which may not always be feasible due to privacy or storage issues. In contrast, our proposed method, PCC, remarkably surpasses all of them across all tasks without storing the original data (but their representations), thus avoiding such privacy issues.

5. CONCLUSIONS

This work studies the goodness of five state-of-the-art CL methods (Replay, EWC, iCaRL, GEM, and L2P) in the context of CIL for music classification. Additionally, we propose a simple yet effective CIL method (PCC) that relies on the generalizability of foundation models.

Our results reveal that current state-of-the-art CL methods suffer from catastrophic forgetting, whereas the proposed approach achieves the best results over four different music classification tasks. This highlights the need to investigate specific CL methods for music classification.

The results obtained with PCC showcase the robustness and utility of the features extracted with foundation models. We can only expect these models to improve over time, managing to extract more generalizable features for a wider range of tasks. This leads us to believe that it is worth further exploring these approaches for CL in music classification.

In future work, we also plan to study more sophisticated strategies for selecting the prototypes in PCC to improve both accuracy and robustness.

6. ACKNOWLEDGEMENTS

This paper is supported by grant CISEJI/2023/9 from “Programa para el apoyo a personas investigadoras con talento (Plan GenT) de la Generalitat Valenciana”.

7. REFERENCES

[1] J. S. Downie, “Music Information Retrieval,” *Annual Review of Information Science and Technology*, vol. 37, no. 1, pp. 295–340, 2003.

[2] Z. Fu, G. Lu, K. M. Ting, and D. Zhang, “A Survey of Audio-Based Music Classification and Annotation,” *IEEE Transactions on Multimedia*, vol. 13, no. 2, pp. 303–319, 2010.

[3] N. Pelchat and C. M. Gelowitz, "Neural Network Music Genre Classification," *Canadian Journal of Electrical and Computer Engineering*, vol. 43, no. 3, pp. 170–173, 2020.

[4] J. Wilkins, P. Seetharaman, A. Wahl, and B. Pardo, "VocalSet: A Singing Voice Dataset," in *Proceedings of the 19th International Society for Music Information Retrieval Conference*, Paris, France, 2018, pp. 468–474.

[5] 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 *Proceedings of the 34th International Conference on Machine Learning - Volume 70*. PMLR, 2017, pp. 1068–1077.

[6] M. Schedl, "Deep Learning in Music Recommendation Systems," *Frontiers in Applied Mathematics and Statistics*, p. 44, 2019.

[7] M. F. McKinney, "Features for Audio and Music Classification," in *Proceedings of the 6th International Society for Music Information Retrieval Conference*, 2003.

[8] M. I. Mandel and D. P. Ellis, "Song-Level Features and Support Vector Machines for Music Classification," in *Proceedings of the 6th International Society for Music Information Retrieval Conference*, 2005.

[9] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Convolutional recurrent neural networks for music classification," in *Proceedings of the 42nd IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2017, pp. 2392–2396.

[10] Y. Li, R. Yuan, G. Zhang, Y. Ma, X. Chen, H. Yin, C. Lin, A. Ragni, E. Benetos, N. Gyenge *et al.*, "MERT: Acoustic Music Understanding Model with Large-Scale Self-supervised Training," *arXiv preprint arXiv:2306.00107*, 2023.

[11] D.-W. Zhou, Q.-W. Wang, Z.-H. Qi, H.-J. Ye, D.-C. Zhan, and Z. Liu, "Deep class-incremental learning: A survey," *arXiv preprint arXiv:2302.03648*, 2023.

[12] M. McCloskey and N. J. Cohen, "Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem," in *Psychology of Learning and Motivation*. Elsevier, 1989, vol. 24, pp. 109–165.

[13] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A Continual Learning Survey: Defying Forgetting in Classification Tasks," *IEEE Rransactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3366–3385, 2021.

[14] J. P. Jeong Choi, Jongpil Lee and J. Nam, "Zero-shot learning for audio-based music classification and tagging," in *Proceedings of the 20th International Society for Music Information Retrieval Conference*. ISMIR, Nov. 2019, pp. 67–74. [Online]. Available: https://doi.org/10.5281/zenodo.3527741

[15] Y. Wang, N. J. Bryan, M. Cartwright, J. Pablo Bello, and J. Salamon, "Few-shot continual learning for audio classification," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 321–325.

[16] D.-W. Zhou, Q.-W. Wang, Z.-H. Qi, H.-J. Ye, D.-C. Zhan, and Z. Liu, "Deep Class-Incremental Learning: A Survey," *arXiv preprint arXiv:2302.03648*, 2023.

[17] J. Spijkervet and J. A. Burgoyne, "Contrastive learning of musical representations," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference*. ISMIR, Oct. 2021, pp. 673–681.

[18] 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.

[19] J. Lee, J. Park, K. L. Kim, and J. Nam, "Samplecnn: End-to-end deep convolutional neural networks using very small filters for music classification," *Applied Sciences*, vol. 8, no. 1, 2018.

[20] R. Ratcliff, "Connectionist models of recognition memory: constraints imposed by learning and forgetting functions," *Psychological Review*, vol. 97, no. 2, p. 285, 1990.

[21] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[22] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "iCaRL: Incremental Classifier and Representation Learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5533–5542.

[23] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.

[24] C. Kereliuk, B. L. Sturm, and J. Larsen, "Deep Learning and Music Adversaries," *IEEE Transactions on Multimedia*, vol. 17, no. 11, pp. 2059–2071, 2015.

[25] Z. Wang, Z. Zhang, C.-Y. Lee, H. Zhang, R. Sun, X. Ren, G. Su, V. Perot, J. Dy, and T. Pfister, "Learning to Prompt for Continual Learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 139–149.