

LEARNABLE LOSS MIXUP FOR SPEECH ENHANCEMENT

Oscar Chang, Dung N. Tran, Kazuhito Koishida

Microsoft Research

ABSTRACT

Mixup [1] is a recently proposed learning paradigm that improves the generalization of deep neural networks by training them on virtual data sampled from linear interpolations of examples and their labels. However, applying it to speech enhancement is challenging, because mixup was not designed for non-classification tasks and its success is contingent on the shape of the mixing distribution. We propose a generalization of mixup that mixes the losses instead of the labels, and automatically learns a non-linear mixing function by conditioning on the mixed data. On the VCTK benchmark, our proposal significantly outperforms standard training, learnable label mixup, and linear loss mixup. It achieves 3.26 PESQ, surpassing the previous state-of-the-art by 6 points.

1. INTRODUCTION

Data augmentation has consistently been found to boost the generalization of deep neural networks across a gamut of different data domains, and is essential to achieving state-of-the-art results in applications as diverse as object recognition [2], image super-resolution [3], sentiment analysis [4], semantic segmentation [5], machine translation [6], and audio classification [7]. It is also indispensable in settings where data is limited or hard to collect, for example in biomedical imaging [8]. However, data augmentation techniques typically involve specialized domain knowledge and are not transferable between different domains.

Unlike standard techniques, *Mixup* is domain-independent and can be generally applied. At each training iteration, mixup creates a virtual example $(\lambda x_1 + (1 - \lambda)x_2, \lambda y_1 + (1 - \lambda)y_2)$ by randomly sampling two labeled training examples $(x_1, y_1), (x_2, y_2)$ and a mixing coefficient $\lambda \in [0, 1]$. When used to train deep neural networks on classification tasks, mixup has been shown to achieve both better generalization and increased model calibration across a range of data domains: images, audio, text, and tabular data [1, 9].

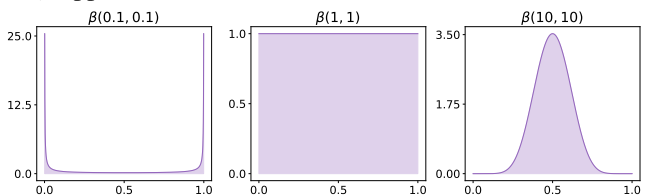
Mixup is not just data augmentation, but also a fundamentally new approach to deep supervised learning. It departs from Empirical Risk Minimization (ERM), which is the long-standing principle of minimizing a loss function on examples in a training set, in favor of Vicinal Risk Minimization, which optimizes a loss function on all points within the *vicinity* of the training set, in this case defined to be points that lie on

the line between any two training examples. As such, despite its wide success, it is still not clear *why* mixup works. It is not difficult to find a counter-example where linearly interpolating data with mixup will not work, for example regressing against $y = x^2$ where the virtual data deviates from the true model.

Even though its formulation requires nothing beyond a supervised learning setting, generalizing mixup beyond classification tasks has been a persistent open problem [1]. It should not be surprising that mixing different labels is helpful in classification tasks, since this can be viewed as a form of label smoothing [10]. Nevertheless, mixed labels may not be a good idea in general, because noisy labels degrade the performance of deep networks [11]. In particular, for the task of speech enhancement, where a deep network is trained to map from a noisy audio sample to its clean equivalent, mixing the labels produces a noisy training target, since the mix of two clean audio samples is a noisy one.

Furthermore, it has been observed that the success of mixup is highly sensitive to the shape of the mixing distribution [12, 9, 13]. The mixup authors [1] proposed for λ to be drawn from a symmetric Beta distribution $\beta(\alpha, \alpha)$, but we can see from Fig. 1 that different values of α result in very different distributions. They found that for ImageNet, large α causes under-fitting and recommended $\alpha \in [0.1, 0.4]$. However, for CIFAR-10, mixup was found to be robust to large α , and $\alpha = 8$ was recommended in the case of noisy labels. On CIFAR-100, [12] found that while $\alpha = 0.4$ outperforms standard ERM training, $\alpha = 0.1$ performs slightly worse and $\alpha = 5$ significantly worse. These findings make it clear that even though mixup is domain independent, the

Fig. 1. PDFs of $\beta(\alpha, \alpha)$ with different α . As $\alpha \rightarrow 0$, β approaches a delta function at 0.0 and 1.0 (mixup becomes ERM). At $\alpha = 1$, β becomes a uniform distribution. As $\alpha \rightarrow \infty$, β approaches a standard normal distribution.



optimal mixing distribution is dataset dependent. Manually tuning the mixing distribution to fit a given dataset is non-trivial, because it involves trial-and-error on multiple training runs, making it much more expensive than ERM. This is a particular challenge for speech applications, where the computational expense is already considerably high.

1.1. Our Contribution

Our work studies the application of mixup to the task of speech enhancement. Mixed labels do not make good training targets, because the mix of two clean samples becomes a noisy sample. We propose an approach inspired by the cocktail party problem, where the input is a mixture of two noisy signals but the model learns to separate the component signals by training against a mixture of two losses, one for each clean sample. We call this approach *Loss Mixup*, as opposed to classical mixup that is label-based. While loss mixup is clearly distinct from label mixup, we prove in Corollary 3.2 that for the widely used case of the cross entropy loss in classification, they are indeed equivalent.

As with label mixup, tuning the mixing distribution for the two loss terms can be expensive. To avoid this, we reparametrize the mixing distribution parameters into loss function variables that can be learned via gradient descent. As opposed to manually tuning a global mixing distribution that is dataset dependent, this key idea, which we term *Learnable Loss Mixup*, conditions the mixing on the features of individual mixed inputs, thus enabling a more fine-grained mixing that is datapoint dependent.

On the VCTK speech enhancement dataset [14], we show that learnable loss mixup significantly outperforms standard ERM training, learnable label mixup, and linear loss mixup. It achieves 3.26 PESQ, surpassing the previous state-of-the-art by 6 points.

The rest of the paper is organized as follows: we briefly survey related work for mixup in Section 2, introduce Learnable Loss Mixup in Section 3, and finally discuss experimental results and their implications in Section 4.

2. RELATED WORK

[7] proposed a mixing method for audio classification that also scales the mixed sample according to the amplitude of the component sound signals. [12] learns to tune the mixing distribution so that it avoids *manifold intrusion*, which is when the interpolation of two datapoints produces virtual data that look like a third class. It is rare to find mixup applied to non-classification tasks in the literature, but [3] notably proposed CutMix for image super-resolution, where mixup is applied to small randomly sampled windows rather than the whole image. To our knowledge, we are the first to propose loss rather than label mixup and apply it to a non-classification task.

Single-channel speech enhancement is typically modeled as a problem of removing additive noise: given a noisy sample $x = s + n$, we want to recover the clean signal s by removing the noisy component n . Historically, non-learning-based approaches like Wiener filtering were used, but they have since been superseded by deep supervised learning methods that train on paired samples of x and s . The deep network can operate directly in the time domain [15, 16], or indirectly in the frequency domain [17, 18, 19, 20], where the audio sample is first processed into a spectrogram with a short-time Fourier Transform (STFT). We follow the latter approach.

3. LEARNABLE LOSS MIXUP

The main difference between ERM, classical mixup, and our proposed version of mixup lies in the training loss, which we list in Table 1. Loss mixup mixes the losses instead of the labels, and learnable mixup shapes the mixing coefficient λ with a mixing function ϕ . This categorization has classical mixup as Linear Label Mixup, and gives rise to three possible variants: Learnable Label Mixup, Linear Loss Mixup, and Learnable Loss Mixup. We show that despite being distinct concepts, both loss and label mixup can be viewed as natural generalizations of the application of classical mixup in the classification setting.

3.1. Relationship Between Label Mixup and Loss Mixup

Theorem 3.1. *The gradient update for linear label mixup is the same as that for linear loss mixup if and only if the gradient of the loss \mathcal{L} is an affine map of the labels y .*

$$\begin{aligned} \text{Proof. } \nabla_{\theta} \mathcal{L}(\hat{y}, \lambda y_1 + (1 - \lambda)y_2) \\ &= \lambda \nabla_{\theta} \mathcal{L}(\hat{y}, y_1) + (1 - \lambda) \nabla_{\theta} \mathcal{L}(\hat{y}, y_2) \\ &= \nabla_{\theta} [\lambda \mathcal{L}(\hat{y}, y_1) + (1 - \lambda) \mathcal{L}(\hat{y}, y_2)], \quad \forall \lambda \in [0, 1]. \end{aligned}$$

Corollary 3.2. *Linear label and linear loss mixup are equivalent for the cross-entropy and the mean squared error loss.*

Corollary 3.2 implies that for most classification and regression tasks, where the cross entropy and mean squared error loss are used respectively, the two generalizations of classical mixup are equivalent. However, this does not hold in general, especially not for more application-specific loss

Table 1. Comparison between different versions of mixup.

Method	Training Loss
ERM	$\mathcal{L}(\hat{y}, y)$
Linear Label Mixup	$\mathcal{L}(\hat{y}, \lambda y_1 + (1 - \lambda)y_2)$
Learnable Label Mixup	$\mathcal{L}(\hat{y}, \phi(\lambda)y_1 + (1 - \phi(\lambda))y_2)$
Linear Loss Mixup	$\lambda \mathcal{L}(\hat{y}, y_1) + (1 - \lambda) \mathcal{L}(\hat{y}, y_2)$
Learnable Loss Mixup	$\phi(\lambda) \mathcal{L}(\hat{y}, y_1) + (1 - \phi(\lambda)) \mathcal{L}(\hat{y}, y_2)$

functions like the log-spectral distance (LSD) that is commonly used in speech. Hence, the choice between label and loss mixup is crucial because it can lead to different model performance for the task of speech enhancement.

Mixed labels in speech enhancement train the model to produce a noisy audio sample, while mixed losses train the model to selectively denoise the mixed data into its constituent clean components. Intuitively, we can see that loss mixup makes more sense as a data augmentation mechanism in this case because it trains the model to implicitly solve an instance of the cocktail party problem.

Nevertheless, loss mixup does not obviate the need to tune hyperparameters for the mixing distribution. In what follows, we discuss how the reparametrization trick can be used to turn these hyperparameters into learnable parameters via gradient descent, i.e. learnable mixup.

3.2. The Reparametrization Trick

It is not possible to directly learn the hyperparameters of the mixing distribution $\beta(\alpha, \alpha)$ via gradient descent, since they only affect the distribution of the sampled losses but do not appear in the loss function itself. But we can transform the loss function so that λ is sampled from a fixed distribution $\mathcal{U}(0, 1)$, but shaped with a corresponding function ϕ that is parametrized by the mixing distribution parameters α . This allows us to calculate Monte Carlo estimates of the loss function that is differentiable with respect to α :

$$\begin{aligned} \nabla_{\alpha} \mathbb{E}_{\lambda \sim \beta(\alpha, \alpha)} \mathcal{L}(\lambda) &= \nabla_{\alpha} \mathbb{E}_{\lambda \sim \mathcal{U}(0, 1)} \mathcal{L}(\phi_{\alpha}(\lambda)) \\ &= \mathbb{E}_{\lambda \sim \mathcal{U}(0, 1)} \nabla_{\alpha} \mathcal{L}(\phi_{\alpha}(\lambda)). \end{aligned} \quad (1)$$

Note that the reparametrization trick does not only apply to the symmetric Beta distribution $\beta(\alpha, \alpha)$. Hence, we are not restricted to the specific ϕ_{α} induced by the symmetric Beta, since any function that is an inverse CDF is a potential candidate for the reparametrization of a mixing distribution. Below, we define the desired properties of a mixing function ϕ , and propose a more suitable parametrization.

3.3. Representation Theorem for Mixing Functions

Definition 3.3. $\phi : [0, 1] \rightarrow [0, 1]$ is said to be a mixing function if

1. $\phi(1) = 1$,
2. $\forall \lambda \in [0, 1] : 1 - \phi(\lambda) = \phi(1 - \lambda)$,
3. and ϕ is monotonically increasing.

The first part of the definition is motivated by how loss mixup should revert to the special case of ERM when there is no mixing being done, i.e. $\lambda = 1$. The second part enforces symmetry: loss mixup should produce the same loss when (x_1, y_1) and (x_2, y_2) are swapped, and λ becomes $1 - \lambda$. The

third part ensures that ϕ is an inverse CDF. Our definition implies a general representation for mixing functions that reduces the parametrization of ϕ to a simpler function ρ .

Theorem 3.4. A mixing function ϕ can be represented by some monotonically increasing ρ that satisfies $\rho(0) = 0$.

$$\phi(\lambda) = \frac{\rho(\lambda)}{\rho(\lambda) + \rho(1 - \lambda)}.$$

Proof. We can verify Definition 3.3 as follows.

$$\phi(1) = \frac{\rho(1)}{\rho(1) + \rho(0)} = 1,$$

$$1 - \phi(\lambda) = \frac{\rho(1 - \lambda)}{\rho(\lambda) + \rho(1 - \lambda)} = \phi(1 - \lambda),$$

$$\text{and } \phi(\lambda) = \frac{1}{1 + \frac{\rho(1 - \lambda)}{\rho(\lambda)}} \text{ is monotonically increasing.}$$

While classical mixup ties the mixing of the inputs with that of the outputs, separating them into distinct mixing functions $\phi_{\text{input}}, \phi_{\text{output}}$ for reparametrized inputs $\phi_{\text{input}}(\lambda)x_1 + (1 - \phi_{\text{input}}(\lambda))x_2$ and outputs $\phi_{\text{output}}(\lambda)\mathcal{L}(\hat{y}, y_1) + (1 - \phi_{\text{output}}(\lambda))\mathcal{L}(\hat{y}, y_2)$ makes the model more expressive. In particular, for cases where loss and label mixup are the same, notice that setting $\phi_{\text{input}} = \phi_{\text{output}}$ biases the model towards linear functions between x and y (this is because all the virtual points lie on the line between (x_1, y_1) and (x_2, y_2)).

We enable non-linear mixing in learnable mixup by fixing ϕ_{input} as the identity function, and parametrizing ϕ_{output} with Eqn. 2. This has the advantage of computational efficiency, since ϕ_{output} can be conditioned on an embedding of the mixed input $\lambda x_1 + (1 - \lambda)x_2$, instead of separately on x_1 and x_2 . It is easy to check that Eqn. 2 satisfies the conditions of Theorem 3.4, and allows ρ the flexibility of being either convex or concave as long as $C > 1$ (which is important for reasons we discuss below).

$$\rho_{\text{output}}(\lambda) = \text{pow} \left(\lambda, C \sigma(\text{MLP}(\text{Embed}(\lambda x_1 + (1 - \lambda)x_2))) \right). \quad (2)$$

3.4. Shape of the Mixing Function

Finally, we visualize how different realizations of ρ affect the shape of the mixing function ϕ in Fig. 3, and explain how they regularize the model. Convex ρ makes ϕ flat near the endpoints, reflecting the principle behind common regularization methods like dropout that small changes in the input should not change the output. Conversely, concave ρ flattens ϕ near the midpoint, which reduces the model's dependence on noisy data by forcing both small and large mixing to result in similar losses. Instead of manually tuning a global mixing distribution that is merely optimal on average for the whole dataset, learnable mixup helps the model automatically learn the optimal ρ , and thus ϕ , for each individual mixed input, combining the advantages of both regularization regimes.

Fig. 2. Spectrograms for a pair of noisy and clean samples, their mix, and enhanced samples produced by loss and label mixup.

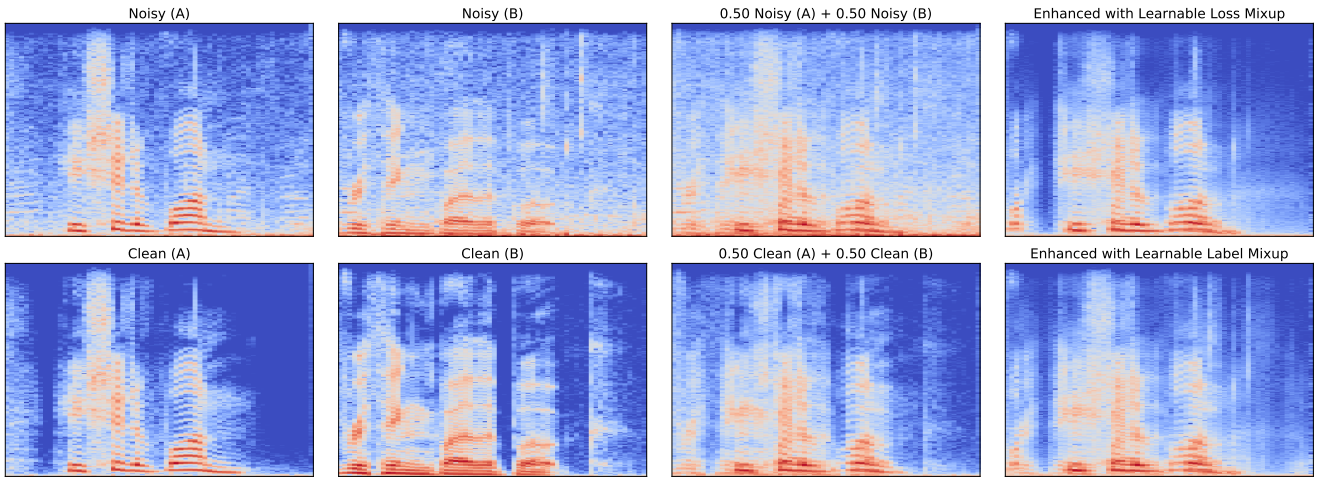


Fig. 3. Shape of mixing function ϕ for different ρ .

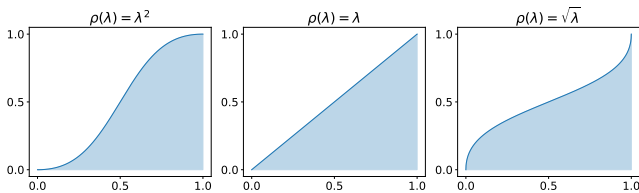


Table 2. Ablation study on VCTK with standard error calculated over 5 random seeds. LLM1: Learnable Label Mixup. LLM2: Linear Loss Mixup. LLM3: Learnable Loss Mixup.

Method	PESQ	CSIG	CBAK	COVL
ERM	3.18 ± 0.00	4.47 ± 0.00	3.52 ± 0.02	3.86 ± 0.00
LLM1	3.10 ± 0.01	4.44 ± 0.00	3.41 ± 0.01	3.80 ± 0.01
LLM2	3.20 ± 0.01	4.48 ± 0.00	3.36 ± 0.01	3.87 ± 0.00
LLM3	3.26 ± 0.01	4.49 ± 0.00	3.27 ± 0.01	3.91 ± 0.00

4. EXPERIMENTAL RESULTS AND DISCUSSION

The VCTK dataset [14] is a popular benchmark for single-channel speech enhancement [20, 16, 17, 18, 19]. The dataset contains 30 speakers (28 training, 2 test), 15 noise types (10 training, 5 test), and 8 signal-to-noise ratios (4 training, 4 test). We randomly split 1% of the training set to form the validation set, and test the models with the highest validation PESQ. We follow standard practice in down-sampling all the audio samples from 48kHz to 16kHz, and process them into spectrograms by applying a 512-point STFT with a Hanning window of size 512 and hop length 256. The last frequency bin is removed to form inputs of size 64 x 256 x 1. We train a Monster UNet [21] on the LSD loss using Adam with learning rate 10^{-4} , $0.5 \beta_1$, $0.9 \beta_2$, $0.1 \ell_2$ -regularization, and batch size 64, for 450 epochs. For ρ_{output} , we set $C = 5$, the UNet’s bottleneck layer as the embedding, and an MLP with one hidden

Table 3. Comparison with prior work on VCTK.

Model	PESQ	CSIG	CBAK	COVL
MMSE-GAN [17]	2.53	3.80	3.12	3.14
D+M [18]	2.73	3.94	3.35	3.33
UNet [19]	2.90	4.22	3.32	3.58
T-GSA [20]	3.06	4.18	3.59	3.62
DEMUCS [15]	3.07	4.31	3.40	3.63
RHR-Net [16]	3.20	4.37	4.02	3.82
Learnable Loss Mixup	3.26	4.49	3.27	3.91

layer of width 512.

We conducted an ablation study on VCTK over 5 random seeds, and list our findings in Table 2. Using ERM as a baseline, we find that learnable loss mixup boosted the performance of our UNet model significantly by 8 points from 3.18 to 3.26 PESQ. Consistent with our earlier discussions, label mixup is not sensible for the task of speech enhancement. Learnable label mixup fared much worse than ERM at 3.10 PESQ, and we stopped an initial run of linear label mixup because it did considerably worse than its learnable counterpart. Linear loss mixup outperformed ERM slightly by 2 points at 3.20 PESQ, demonstrating the benefits of loss mixup even when the mixing is not optimal. Interestingly, while learnable loss mixup improved overall speech quality (PESQ and COVL) and the clean signal (CSIG), it degraded that of the background signal (CBAK). Fig. 2 shows that loss mixup generates more white noise than label mixup, effectively “smoothing” out the background. This is consistent with our cocktail party interpretation, where the model is trained to focus on the clean signal and learns to disregard the background.

Our work surpassed the previous state-of-the-art considerably by 6 PESQ points (Table 3), and we aim to explore its application to other tasks in future.

5. REFERENCES

- [1] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz, “mixup: Beyond empirical risk minimization,” *arXiv preprint arXiv:1710.09412*, 2017.
- [2] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le, “Autoaugment: Learning augmentation policies from data,” *arXiv preprint arXiv:1805.09501*, 2018.
- [3] Jaejun Yoo, Namhyuk Ahn, and Kyung-Ah Sohn, “Rethinking data augmentation for image super-resolution: A comprehensive analysis and a new strategy,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8375–8384.
- [4] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le, “Unsupervised data augmentation for consistency training,” *arXiv preprint arXiv:1904.12848*, 2019.
- [5] Andrew Tao, Karan Sapra, and Bryan Catanzaro, “Hierarchical multi-scale attention for semantic segmentation,” *arXiv preprint arXiv:2005.10821*, 2020.
- [6] Sergey Edunov, Myle Ott, Michael Auli, and David Grangier, “Understanding back-translation at scale,” *arXiv preprint arXiv:1808.09381*, 2018.
- [7] Yuji Tokozume, Yoshitaka Ushiku, and Tatsuya Harada, “Learning from between-class examples for deep sound recognition,” in *International Conference on Learning Representations*, 2018.
- [8] Maayan Frid-Adar, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan, “Synthetic data augmentation using gan for improved liver lesion classification,” in *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*. IEEE, 2018, pp. 289–293.
- [9] Sunil Thulasidasan, Gopinath Chennupati, Jeff A Bilmes, Tanmoy Bhattacharya, and Sarah Michalak, “On mixup training: Improved calibration and predictive uncertainty for deep neural networks,” in *Advances in Neural Information Processing Systems*, 2019, pp. 13888–13899.
- [10] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna, “Rethinking the inception architecture for computer vision,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.
- [11] Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa, “Joint optimization framework for learning with noisy labels,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5552–5560.
- [12] Hongyu Guo, Yongyi Mao, and Richong Zhang, “Mixup as locally linear out-of-manifold regularization,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, vol. 33, pp. 3714–3722.
- [13] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio, “Manifold mixup: Better representations by interpolating hidden states,” in *International Conference on Machine Learning*, 2019, pp. 6438–6447.
- [14] Cassia Valentini-Botinhao, Xin Wang, Shinji Takaki, and Junichi Yamagishi, “Speech enhancement for a noise-robust text-to-speech synthesis system using deep recurrent neural networks,” 2016.
- [15] Alexandre Defossez, Gabriel Synnaeve, and Yossi Adi, “Real time speech enhancement in the waveform domain,” *arXiv preprint arXiv:2006.12847*, 2020.
- [16] Jalal Abdulbaqi, Yue Gu, Shuhong Chen, and Ivan Marsic, “Residual recurrent neural network for speech enhancement,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6659–6663.
- [17] Meet H Soni, Neil Shah, and Hemant A Patil, “Time-frequency masking-based speech enhancement using generative adversarial network,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5039–5043.
- [18] Jian Yao and Ahmad Al-Dahle, “Coarse-to-fine optimization for speech enhancement,” *arXiv preprint arXiv:1908.08044*, 2019.
- [19] Ahmet E Bulut and Kazuhito Koishida, “Low-latency single channel speech enhancement using u-net convolutional neural networks,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6214–6218.
- [20] Jaeyoung Kim, Mostafa El-Khamy, and Jungwon Lee, “T-gsa: Transformer with gaussian-weighted self-attention for speech enhancement,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6649–6653.
- [21] Oscar Chang, “Monster unet architecture for speech enhancement,” Sept. 2020.