INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Single-channel speech enhancement using learnable loss mixup

Oscar Chang[1]*, *Dung N. Tran*[2], *Kazuhito Koishida*[2]

[1]Columbia University, USA
[2]Microsoft Corporation, USA

oscar.chang@columbia.edu, {dung.tran, kazukoi}@microsoft.com

Abstract

Generalization remains a major problem in supervised learning of single-channel speech enhancement. In this work, we propose *learnable loss mixup (LLM)*, a simple and effortless training diagram, to improve the generalization of deep learning-based speech enhancement models. *Loss mixup*, of which *learnable loss mixup* is a special variant, optimizes a mixture of the loss functions of random sample pairs to train a model on virtual training data constructed from these pairs of samples. In *learnable loss mixup*, by conditioning on the mixed data, the loss functions are mixed using a non-linear mixing function automatically learned via neural parameterization. Our experimental results on the VCTK benchmark show that *learnable loss mixup* achieves 3.26 PESQ, outperforming the state-of-the-art.

Index Terms: speech enhancement, noise reduction, deep neural network, convolutional neural network, supervised learning, regression, mixup, loss mixup, learnable loss mixup

1. Introduction

Speech enhancement, the problem of extracting a clean speech signal from its noisy version, plays a crucial role in speech applications and has been extensively investigated in the literature. Classical approaches employ signal processing algorithms [1, 2, 3, 4] and produce satisfactory results in stationary noise environments. However, they fail in more challenging scenarios, such as nonstationary noise, and have been surpassed by deep supervised learning methods [5, 6, 7, 8, 9, 10, 11]. Deep supervised learning for speech enhancement uses pairs of clean and noisy speech signals to train an expressive deep neural network (DNN) able to robustly predict clean speech in highly non-stationary noise environments. They have become the workhorse for modern speech enhancement systems.

State-of-the-art supervised learning-based speech enhancement in the literature typically trains the model by solving an *Empirical Risk Minimization (ERM)* problem [12]. This approach, however, has a major drawback of learning the network behavior only on the training samples. This limitation leads to undesirable behavior of the model beyond the training set. A straightforward and common remedy is to extend the training set which, unfortunately, is expensive and laborious to acquire and label in speech applications. Another common approach is data augmentation which, however, is domain-dependent and thus requires expert knowledge. Improving generalization, the network's ability to produce robust predictions in unseen noise environments, thus remains a major challenge in speech enhancement.

Our contribution. To tackle this problem, we introduce *loss mixup (LM)* with a special variant called *learnable loss mixup (LLM)*. During training, *loss mixup* constructs virtual samples by mixing random noisy training samples. It, then, trains the network on the augmented data by minimizing a mixture of the loss functions of the individual mixing samples. In essence, *loss mixup* is not only a data augmentation method but also an instance of Vicinal Risk Minimization (VRM) [13], a training regime optimizing the loss function within the vicinity of the training set, suitable for regression problems such as speech enhancement. The advantages of *loss mixup* are threefold. First, it augments the training data with richer noise and interference conditions via virtual noisy samples. This data augmentation procedure is domain-independent and can be implemented in a few lines of code without expert knowledge. Moreover, by monitoring the network in the vicinity of training samples, *loss mixup* encourages smooth regression curves and thus stabilizes the network behavior outside the training set. Finally, optimizing the loss mixture instead of the target mixture retains the individual clean targets of the mixing noisy samples, avoiding the degradation of the network in regression settings [14].

Manually tuning the mixing parameter, sampled from a mixing distribution in our framework, for the loss mixture is nontrivial. Therefore, we propose *learnable loss mixup* to automatically learn the mixing distribution via gradient descent by reparameterizing the mixing distribution parameters into loss function variables. Furthermore, we introduce conditions allowing reparameterization by neural networks which can be trained efficiently via backpropagation. On the VCTK speech enhancement dataset [15], we show that *learnable loss mixup* significantly outperforms standard ERM training. It achieves 3.26 PESQ, surpassing the previous state-of-the-art by 0.06 points.

Related work. Our method resembles the sample mixing mechanism of *mixup* [16], a new training method aiming to improve the generalization of supervised learning for classification problems. Similar to our *loss mixup*, mixup, which we call *label mixup* in this paper, adopts the Vicinal Risk Minimization approach by generating virtual samples from random training samples. However, unlike *loss mixup*, label mixup trains the network on virtual targets obtained from the labels of the respective mixing samples. As label mixing is a form of label smoothing [17], label mixup is beneficial to classification tasks and has achieved both better generalization and increased model calibration in classification applications across various data domains such as images, audio, text, and tabular data [16, 18]. Nevertheless, applying label mixup to regression problems, such as speech enhancement, is inappropriate since mixing clean targets produces noisy labels which degrade the network performance in regression settings [14]. As mentioned above, *loss mixup* overcomes this issue by retaining the targets of the mixing samples and optimizing the mixture of the respective individual loss functions. This insight is supported by our ablation study, in the experiment section, confirming the supe-

* work done during internship at Microsoft.

http://dx.doi.org/10.21437/Interspeech.2021-859

riority of *loss mixup* over *label mixup* in speech enhancement. To our knowledge, we are the first to propose the sample mixup principle beyond non-classification tasks.

Outline. The rest of the paper is organized as follows. We introduce *loss mixup* in Section 2 and draw connection between *loss mixup* and *label mixup*. In Section 3, we present *learnable loss mixup* by detailing the reparameterization trick and conditions allowing an efficient neural parameterization of the mixing function. We discuss experimental results and their implications in Section 4, and, finally, conclude our work in Section 5.

2. Loss mixup

We seek an enhanced version of an observed noisy speech signal $\boldsymbol{x}$ via a learning-based approach. As commonly the case in practice, the noisy speech is assumed to follow an additive noise model:

$$\boldsymbol{x} = \boldsymbol{s} + \boldsymbol{n}, \tag{1}$$

where $\boldsymbol{s}$ and $\boldsymbol{n}$ are the unknown clean speech and the unknown additive noise, respectively. We train a deep neural network (DNN) f to extract the clean speech from the noisy speech without noise information.

We adopt a supervised learning regime that trains the DNN on a dataset $\mathcal{D} = \{(\boldsymbol{x}_j, \boldsymbol{s}_j)\}_j^M$ of M pairs of noisy and clean speech. Let ℓ be a loss function enforcing the consistency between the enhanced speech and the corresponding clean speech, current state-of-the-art methods typically train the network by solving the *empirical risk minimization (ERM)* problem:

$$\min \frac{1}{M} \sum_{j=1}^{M} \ell\big(f\left(\boldsymbol{x}_j\right), \boldsymbol{s}_j\big), \tag{2}$$

where $f\left(\boldsymbol{x}_j\right)$ is the prediction for noisy sample $\boldsymbol{x}_j$. Despite being the workhorse of modern speech enhancement systems, ERM possesses a major drawback: It causes the network to memorize the training data by limiting the training on a finite set of samples. Data memorization, in turn, leads to poor network generalization.

2.1. Loss mixup training

To alleviate the generalization issue, we propose *loss mixup (LM)* training, a generalized version of *mixup* [16], for speech enhancement. Given two random training samples $(\boldsymbol{x}_j, \boldsymbol{s}_j)$ and $(\boldsymbol{x}_i, \boldsymbol{s}_i)$, and a mixing parameter $\lambda \in [0, 1]$, *loss mixup* constructs virtual noisy samples

$$\tilde{\boldsymbol{x}} = \lambda \boldsymbol{x}_j + (1-\lambda)\boldsymbol{x}_k \tag{3}$$

and trains the network by minimizing

$$\mathcal{L}_{LM} = \lambda \ell\big(f\left(\tilde{\boldsymbol{x}}\right), \boldsymbol{s}_j\big) + (1-\lambda)\ell\big(f\left(\tilde{\boldsymbol{x}}\right), \boldsymbol{s}_k\big). \tag{4}$$

Importantly, the virtual noisy samples are constructed during the training stage only. During inference, for a noisy speech signal $\boldsymbol{x}$, the enhanced speech is obtained from the network's output $f(\boldsymbol{x})$.

Similar to label mixup, the success of *loss mixup* is highly sensitive to the mixing parameter λ [19, 18, 20]. For instance, Zhang et. al. [16] proposed for λ to be drawn from a symmetric Beta distribution $\beta(\alpha, \alpha)$ which varies dramatically for different values of α as demonstrated in Fig. 1. Moreover, several studies [16, 19] indicate that the optimal mixing distribution is dataset dependent, and manually tuning the mixing distribution to fit a given dataset is expensive. This challenge motivates our *learnable loss mixup* variant in Section 3.

2.2. Connection to label mixup

Before introducing *learnable loss mixup* in the next section, here, we draw a connection between *loss mixup* and label mixup and highlight that applying them to speech enhancement yields different model performance.

Theorem 2.1. *The gradient update for the label mixup loss, defined as*

$$\ell\big(f\left(\tilde{\boldsymbol{x}}\right), \lambda \boldsymbol{s}_j + (1-\lambda)\boldsymbol{s}_k\big), \tag{5}$$

is identical to that for the loss mixup loss $\mathcal{L}_{LM}$ in (4) if and only if the gradient of the loss ℓ is an affine map of the targets.

Proof. Let θ be the weight of the network. For all $\lambda \in [0, 1]$, $\nabla_\theta \ell\big(f\left(\tilde{\boldsymbol{x}}\right), \lambda \boldsymbol{s}_j + (1-\lambda)\boldsymbol{s}_k\big) = \lambda \nabla_\theta \mathcal{L}\big(f\left(\tilde{\boldsymbol{x}}\right), \boldsymbol{s}_j\big) + (1-\lambda)\nabla_\theta \ell\big(f\left(\tilde{\boldsymbol{x}}\right), \boldsymbol{s}_k\big) = \nabla_\theta\big[\lambda \ell\big(f\left(\tilde{\boldsymbol{x}}\right), \boldsymbol{s}_j\big) + (1-\lambda)\ell\big(f\left(\tilde{\boldsymbol{x}}\right), \boldsymbol{s}_k\big)\big]$ which concludes the proof for the theorem. □

Corollary 2.1.1. *Loss mixup (4) and label mixup (5) are equivalent for the cross-entropy and the mean squared error loss.*

Corollary 2.1.1 implies that *loss mixup* and label mixup are equivalent for classification and regression tasks utilizing the cross-entropy and mean squared error loss. However, for application-specific loss functions such as the log-spectral distance (LSD), a common loss function in speech applications, *loss mixup* and label mixup lead to different model performance for speech enhancement. Our ablation study in Section 4 confirms this insight.

3. Learnable loss mixup

3.1. The reparameterization trick

We follow the approach in [16] that samples λ from a mixing distribution $p(\alpha)$. As discussed in the previous section, manually tuning the mixing distribution is impractical. Furthermore, directly learning the hyperparameters α of the mixing distribution is unfeasible since they are absent from the loss function. Therefore, we reparameterize the mixed loss function (4) by sampling λ from the uniform distribution $\mathcal{U}(0, 1)$ and shaping λ with a parameterized mixing function ϕ:

$$\mathcal{L}_{LLM} = \phi\left(\lambda\right) \ell\big(f\left(\tilde{\boldsymbol{x}}\right), \boldsymbol{s}_j\big) + (1-\phi\left(\lambda\right))\ell\big(f\left(\tilde{\boldsymbol{x}}\right), \boldsymbol{s}_k\big). \tag{6}$$

Learnable loss mixup (LLM), which trains the network by minimizing (6) instead of (4), allows using gradient descent to learn the hyperparameters of the mixing distribution. In particular, assume λ follows an arbitrary distribution $p(\alpha)$ and let ϕ_α be the inverse cumulative distribution function (CDF) of $p(\alpha)$, the reparameterization trick enables us to calculate Monte

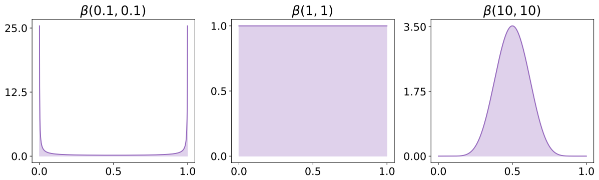

Figure 1: *PDFs of $\beta(\alpha, \alpha)$ with different α. As $\alpha \to 0$, β approaches a delta function at* 0.0 *and* 1.0 *(mixup becomes ERM). At $\alpha = 1$, β becomes a uniform distribution. As $\alpha \to \infty$, β approaches a standard normal distribution.*

Carlo estimates of the loss function differentiable with respect to (w.r.t.) α:

$$\begin{aligned}\nabla_\alpha \mathbb{E}_{\lambda \sim p(\alpha)} \mathcal{L}(\lambda) &= \nabla_\alpha \mathbb{E}_{\lambda \sim \mathcal{U}(0,1)} \mathcal{L}(\phi_\alpha(\lambda)) \\ &= \mathbb{E}_{\lambda \sim \mathcal{U}(0,1)} \nabla_\alpha \mathcal{L}(\phi_\alpha(\lambda)).\end{aligned} \tag{7}$$

The optimal mixing distribution $p(\alpha)$, then, is obtained by minimizing the last term of (7) w.r.t. α via gradient descent.

Eq. (7) implies that any inverse CDF is a candidate for the reparametrization of a mixing distribution. This insight motivates us to define below desired properties of the mixing function ϕ. They allow ϕ to be parameterized by neural networks which can be trained efficiently via backpropagation.

3.2. Representation theorem for the mixing function

Definition 3.1. $\phi : [0, 1] \rightarrow [0, 1]$ *is 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 implies *loss mixup* reverts to ERM when $\lambda = 1$. The second part enforces symmetry: *loss mixup* produces the same loss when $(\boldsymbol{x}_j, \boldsymbol{s}_j)$ and $(\boldsymbol{x}_i, \boldsymbol{s}_i)$ are switched 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 3.1. *A mixing function ϕ can be represented by some monotonically increasing function ρ that satisfies $\rho(0) = 0$:*

$$\phi(\lambda) = \frac{\rho(\lambda)}{\rho(\lambda) + \rho(1 - \lambda)}. \tag{8}$$

Proof. As $\phi(1) = \frac{\rho(1)}{\rho(1)+\rho(0)} = 1$,$\phi(1-\lambda) = \frac{\rho(1-\lambda)}{\rho(\lambda)+\rho(1-\lambda)} = 1 - \phi(\lambda)$, and $\phi(\lambda) = \frac{1}{1+\frac{\rho(1-\lambda)}{\rho(\lambda)}}$ is monotonically increasing, Eq. (8) satisfies Definition 3.1. This concludes the proof for the theorem. □

Note that the reparameterization form (8) of ϕ produces a regularization effect given certain choices of ρ. Specifically, Fig. 2 shows the shapes of the mixing function for various ρ satisfying Theorem 3.1. Convex ρ flattens ϕ near the endpoints, reflecting the principle behind common regularization methods, such as dropout, that small changes in the input preserve 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 produce similar losses. As oppose to manually tuning a global mixing distribution that is optimal on average for the whole dataset, *learnable loss mixup* automatically learns the optimal ρ, and thus ϕ, for each mixed input, combining the advantages of both regularization regimes.

Importantly, Theorem 3.1 permits flexible neural parameterization of the mixing function ϕ. Below, we introduce an instance of such parameterization allowing efficient learning of the mixing function.

3.3. Neural parameterization of the mixing function

We propose a computational efficent parameterization of ϕ by conditioning on the virtual noisy samples. In particular, given

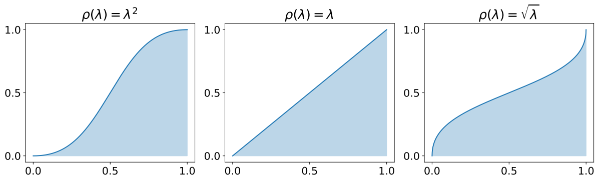

Figure 2: *Shape of mixing function ϕ for different ρ.*

a virtual samples $\tilde{\boldsymbol{x}}$ as in (3), assume our speech enhancement network produces an embedding embed $(\tilde{\boldsymbol{x}})$, we use

$$\rho(\lambda) = \lambda^{C\sigma(\text{MLP}(\text{embed}(\tilde{\boldsymbol{x}})))} \tag{9}$$

to parameterize ϕ, where $C > 1$, σ is the sigmoid function, and MLP is a Multilayer Perception. An example of speech enhancement networks producing such embedding embed $(\tilde{\boldsymbol{x}})$ is the UNet described in Table 1.

The function in (9) satisfies Theorem 3.1 and enables computational efficiency via backpropagation as ϕ in this case is conditioned on an embedding of the mixed samples. Furthermore, when $C > 1$, ρ is Eq. (9) is either convex or concave and thus, following the insight in the previous subsection, regularizes the network to prevent overfitting.

4. Experimental results

In this section, we empirically evaluate and benchmark *learnable loss mixup* against state-of-the-art speech enhancement methods on a public speech enhancement dataset.

4.1. Dataset

The VCTK dataset [15] is a popular benchmark for single-channel speech enhancement [21, 22, 23, 24, 25, 26]. It 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 for validation, and evaluate the models on the test set. The noise types and speakers in the test set are different from those in the training data.

4.2. Objective metrics

We use four popular objective measures for speech enhancement to evaluate the enhanced speech: CSIG, CBAK, COVL, and PESQ. They are computed by comparing the enhanced speech with the corresponding clean reference of each test sample. The first three metrics predict the Mean Opinion Score (MOS) that would result from human perceptual trials: CSIG predicts the signal distortion MOS; CBAK produces the MOS of background-noise intrusiveness; and COVL computes the MOS of the overall signal quality. They produce MOS values from 1 to 5. The last metric, PESQ standing for Perceptual Evaluation of Speech Quality, is a broadly used objective measure for speech quality. Its score is in the range between -0.5 to 4.5. For all of the metrics, the higher value corresponds to the better quality of the enhanced speech. In this work, we focus on PESQ as it is most perceptually consistent with our listening test.

4.3. Experimental setup

We predict the log power spectrum of clean speech from that of the corresponding noisy signal and combine the prediction

Table 1: *UNet Architecture*

Type	Kernel	Stride	Output
Input			$64 \times 256 \times 1$
Conv2d	7×7	1×1	$64 \times 256 \times 128$
Conv2d	5×5	1×2	$64 \times 128 \times 128$
Conv2d	5×5	1×1	$64 \times 128 \times 128$
Conv2d	5×5	1×2	$64 \times 64 \times 128$
Conv2d	5×5	1×1	$64 \times 64 \times 128$
Conv2d	3×3	1×2	$64 \times 32 \times 128$
Conv2d	3×3	1×1	$64 \times 32 \times 128$
⋮	⋮	⋮	(Freq. downsampling)
Conv2d	3×3	1×1	$64 \times 4 \times 128$
Conv2d	3×3	1×2	$64 \times 2 \times 128$
Conv2d	3×2	1×1	$64 \times 2 \times 128$
Conv2d	3×2	1×2	$64 \times 1 \times 128$
Conv2d	3×1	1×1	$64 \times 1 \times 128$
Conv2d	3×1	2×1	$32 \times 1 \times 256$
⋮	⋮	⋮	(Time downsampling)
Conv2d	3×1	1×1	$4 \times 1 \times 256$
Conv2d	3×1	2×1	$2 \times 1 \times 256$
Conv2d	2×1	1×1	$2 \times 1 \times 256$
Conv2d	2×1	2×1	$1 \times 1 \times 512$
SubpixelConv2d	1×1	2×1	$2 \times 1 \times 256$
Conv2d	2×1	1×1	$2 \times 1 \times 256$
SubpixelConv2d	2×1	2×1	$4 \times 1 \times 256$
Conv2d	3×1	1×1	$4 \times 1 \times 256$
SubpixelConv2d	3×1	2×1	$8 \times 1 \times 256$
Conv2d	3×1	1×1	$8 \times 1 \times 256$
⋮	⋮	⋮	(Time upsampling)
SubpixelConv2d	3×1	2×1	$64 \times 1 \times 128$
Conv2d	3×1	1×1	$64 \times 1 \times 128$
SubpixelConv2d	3×1	1×2	$64 \times 2 \times 128$
Conv2d	3×2	1×1	$64 \times 2 \times 128$
SubpixelConv2d	3×2	1×2	$64 \times 4 \times 128$
Conv2d	3×3	1×1	$64 \times 4 \times 128$
SubpixelConv2d	3×3	1×2	$64 \times 8 \times 128$
Conv2d	3×3	1×1	$64 \times 8 \times 128$
⋮	⋮	⋮	(Freq. upsampling)
SubpixelConv2d	3×3	1×2	$64 \times 64 \times 128$
Conv2d	5×5	1×1	$64 \times 64 \times 128$
SubpixelConv2d	5×5	1×2	$64 \times 128 \times 128$
Conv2d	5×5	1×1	$64 \times 128 \times 128$
SubpixelConv2d	5×5	1×2	$64 \times 256 \times 128$
Conv2d	7×7	1×1	$64 \times 256 \times 1$

with the noisy phase to produce the enhanced speech. To obtain the spectrograms, we downsample the data to 16kHz, and apply a 512-point STFT with a Hanning window of size 512 and hop length 256. The last frequency bin is removed to form spectrogram inputs of size 64 x 256 x 1. We train a UNet on the log-spectral distance (LSD) loss [26] using the Adam optimizer with learning rate 10^{-4}, $\beta_1 = 0.5$, $\beta_2 = 0.9$, for 450 epochs. In addition, we use batch size 64 and apply 0.1 ℓ_2-regularization. The UNet, summarized in Table 1, has 56 layers with $20,322,818$ parameters in total. Each decoder layer operates on a concatenation of the preceding layer and a skip connection from its corresponding layer in the encoder. Every convolution layer in the UNet, except the last, is followed by batch normalization, dropout with 0.1 drop rate, and Leaky ReLU with 0.2 negative slope. For ρ in (9), we set $C = 5$, the UNet's bottleneck layer as the embedding, and an MLP with one hidden layer of width 512.

For mixup methods, the virtual noisy samples are generated during the training process only. To compute the metrics on the test set, for each test sample ($\boldsymbol{x}$, $\boldsymbol{s}$), we compare the prediction $f(\boldsymbol{x})$ against the target speech $\boldsymbol{s}$. The final metric on the test set is the average of the metrics of the test samples.

4.4. Results

We benchmarked our proposed framework against classical and state-of-the-art speech enhancement methods including MMSE-GAN [23], D+M [24], UNet [25], Affinity Minimization [26], T-GSA [21], DEMUCS [27], and RHR-Net [22]. Table 2 shows the numerical results of the aforementioned objective metrics on the test dataset for all benchmarked frameworks. For reference, we also report the objective measures computed for the noisy test signals. The results indicated that *learnable loss mixup* outperformed the benchmarked methods in all perceptual metrics excepts CBAK. In particular, it surpassed the state-of-the-art by at least 0.06 in PESQ.

Table 2: *Speech enhancement benchmark*

	PESQ	CSIG	CBAK	COVL
Noisy	1.97	3.35	2.44	2.63
MMSE-GAN [23]	2.53	3.80	3.12	3.14
D+M [24]	2.73	3.94	3.35	3.33
UNet [25]	2.90	4.22	3.32	3.58
UNet-AM [26]	3.04	4.30	3.42	3.69
T-GSA [21]	3.06	4.18	3.59	3.62
DEMUCS [27]	3.07	4.31	3.40	3.63
RHR-Net [22]	3.20	4.37	**4.02**	3.82
Learnable Loss Mixup	**3.26**	**4.49**	3.27	**3.91**

Table 3: *Ablation study on VCTK with standard error calculated over 5 random seeds. LLM1: Learnable Label Mixup. LM: Loss Mixup. LLM2: Learnable Loss Mixup.*

	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
LM	3.20 ± 0.01	4.48 ± 0.00	3.36 ± 0.01	3.87 ± 0.00
LLM2	**3.26** ± 0.01	**4.49** ± 0.00	3.27 ± 0.01	**3.91** ± 0.00

We further conducted an ablation study on VCTK over 5 random seeds, and list our findings in Table 3. Here, learnable label mixup refers to label mixup in which the mixup parameter λ in (5) is substituted by the mixup function ϕ parameterized by (9). Using ERM as a baseline, we found that *learnable loss mixup* , favoring VRM, boosted the performance of our UNet model significantly from 3.18 to 3.26 PESQ. Furthermore, consistent with our earlier discussions, label mixup, despite adopting VRM, is inappropriate for speech enhancement: Learnable label mixup fared significantly worse than ERM at 3.10 PESQ. Finally, *loss mixup* surpassed ERM slightly by 0.02 points at 3.20 PESQ, demonstrating the benefits of *loss mixup* even when the mixing is non-optimal.

5. Conclusions

We presented *loss mixup*, a training framework resembling data augmentation and Vicinal Risk Minimization, to improve the generalization of deep supervised learning for speech enhancement. Furthermore, we derived neural parameterization of the mixing function, resulting in *learnable loss mixup*, to efficiently learn the optimal mixing function via gradient descent. Experimental results showed that our framework outperforms state-of-the-art speech enhancement methods in the literature. Our results suggested that *loss mixup* is a promising approach to boost the generalization of supervised learning for regression problems.

6. References

[1] S. F. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 27, no. 2, pp. 113–120, 1929.

[2] J. S. Lim and A. V. Oppenheim, "All-pole modeling of degraded speech," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 26, no. 3, pp. 197–210, 1978.

[3] ——, "Enhancement and bandwidth compression of noisy speech," *Proceedings of the IEEE*, vol. 67, no. 12, pp. 1586–1604, 1979.

[4] Y. Ephraimand and D. Malah, "Speech enhancement using a minimum mean-square error log-spectral amplitude estimator," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 33, no. 2, pp. 443–445, 1985.

[5] S. Pascual, A. Bonafonte, and J. Serra, "Segan: Speech enhancement generative adversarial network," *Interspeech*, 2017.

[6] D. S. Williamson and D. Wang, "Time-frequency masking in the complex domain for speech dereverberation and denoising," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 7, 2017.

[7] D. Rethage, J. Pons, and X. Serra, "A wavenet for speech denoising," *ICASSP*, 2018.

[8] F. G. Germain, Q. Chen, and V. Koltun, "Speech denoising with deep feature losses," *Arxiv*, 2018.

[9] M. H. Soni, N. Shah, and H. A. Patil, "Time-frequency masking-based speech enhancement using generative adversarial network," *ICASSP*, 2018.

[10] K. Tan and D. Wang, "A convolutional recurrent neural network for real-time speech enhancement," *Interspeech*, 2018.

[11] A. E. Bulut and K. Koishida, "Low-latency single channel speech enhancement using u-net convolutional neural networks," *ICASSP*, 2020.

[12] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*. The MIT Press: The MIT Press, 2012.

[13] O. Chapelle, J. Weston, L. Bottou, and V. Vapnik, "Vicinal risk minimization," in *NIPS*, 2020.

[14] D. Tanaka, D. Ikami, T. Yamasaki, and K. 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.

[15] C. Valentini-Botinhao, X. Wang, S. Takaki, and J. Yamagishi, "Speech enhancement for a noise-robust text-to-speech synthesis system using deep recurrent neural networks." 2016.

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

[17] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[18] S. Thulasidasan, G. Chennupati, J. A. Bilmes, T. Bhattacharya, and S. Michalak, "On mixup training: Improved calibration and predictive uncertainty for deep neural networks," in *Advances in Neural Information Processing Systems*, 2019, pp. 13 888–13 899.

[19] H. Guo, Y. Mao, and R. Zhang, "Mixup as locally linear out-of-manifold regularization," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 3714–3722.

[20] V. Verma, A. Lamb, C. Beckham, A. Najafi, I. Mitliagkas, D. Lopez-Paz, and Y. Bengio, "Manifold mixup: Better representations by interpolating hidden states," in *International Conference on Machine Learning*, 2019, pp. 6438–6447.

[21] J. Kim, M. El-Khamy, and J. 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.

[22] J. Abdulbaqi, Y. Gu, S. Chen, and I. 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.

[23] M. H. Soni, N. Shah, and H. 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.

[24] J. Yao and A. Al-Dahle, "Coarse-to-fine optimization for speech enhancement," *arXiv preprint arXiv:1908.08044*, 2019.

[25] A. E. Bulut and K. 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.

[26] D. Tran and K. Koishida, "Single-channel speech enhancement by subspace affinity minimization," in *INTERSPEECH 2020*. IEEE, 2020.

[27] A. Defossez, G. Synnaeve, and Y. Adi, "Real time speech enhancement in the waveform domain," *arXiv preprint arXiv:2006.12847*, 2020.