# Mixup Augmentation for Generalizable Speech Separation

Ashish Alex
*Centre for Intelligent Sensing*
*Queen Mary U. of London*
London, UK
a.alex@qmul.ac.uk

Lin Wang
*Centre for Intelligent Sensing*
*Queen Mary U. of London*
London, UK
lin.wang@qmul.ac.uk

Paolo Gastaldo
*DITEN*
*U. of Genoa*
Genoa, Italy
paolo.gastaldo@unige.it

Andrea Cavallaro
*Centre for Intelligent Sensing*
*Queen Mary U. of London*
London, UK
a.cavallaro@qmul.ac.uk

***Abstract*—Deep learning has advanced the state of the art of single-channel speech separation. However, separation models may overfit the training data and generalization across datasets is still an open problem in real-world conditions with noise. In this paper we address the generalization problem with Mixup as data augmentation approach. Mixup creates new training examples from linear combinations of samples during mini-batch training. We propose four variations of Mixup and assess the improved generalization of a speech separation model, DPRNN, with cross-corpus evaluation on LibriMix, TIMIT and VCTK datasets. DPRNN allows efficient modelling of longer input sequences by splitting the learnt representation from input mixture segment into small chunks and performing intra and inter chunk operations iteratively. We show that training DPRNN with the proposed Data-only Mixup augmentation variation improves performance on an unseen dataset in noisy conditions when compared to the baseline SpecAugment augmented models, while having comparable performance on the source dataset.**

***Index Terms*—Speech separation, Speech enhancement, Domain generalization.**

## I. Introduction

Speech separation is the task of separating two or more overlapping speech utterances from a mixed speech signal with multiple speakers talking at the same time. Mixed speech often co-exists with environmental noise that deteriorates separation performance. A robust separation model would benefit applications such as automatic speech recognition, hearing aids and voice assistants.

In comparison to multi-channel approaches that exploit the spatial information of sound sources [1], single-channel speech separation is a more challenging task. Deep neural network (DNN) based algorithms are at the forefront for single-channel speech separation [2]–[8], with time-domain end-to-end architectures [4]–[8] outperforming frequency-domain methods [2], [3]. Most time-domain separation models follow an encoder-masker-decoder architecture. The encoder learns a representation from a mixed speech waveform followed by a masker network which learns individual masks for each source in the mixture. Finally, the decoder outputs the individual waveforms for each source in the mixture. While promising results have been reported with recent approaches [5]–[7], the performance of separation models typically drops when tested in real-world and noisy conditions [9], [10], where the noise in the mixture is different from the training dataset [11], [12].

A model is deemed to have good generalizability if similar performance is obtained when tested on data outside of the training data distribution [12]. Lack of generalization typically stems from overfitting the model on the training dataset. Overfitting of separation models can also be alleviated with regularization techniques such as dropout [13], early stopping, weight decay, batch normalization [14], [15]. However, separation models that employ one or more of these regularization strategies underperform with new test subsets outside of their training distribution [12].

Apart from introducing complex architectural changes, the generalization of separation models in noisy real-world conditions can be enhanced by data augmentation. Park et al. [16] proposed SpecAugment augmentation to attenuate the overfitting problem in automatic speech recognition by masking out random consecutive time and frequency bins from the spectral representation. Niel et al. [17] used Mixup augmentation to the intermediate representation in the separation model. A Mixup based strategy was employed in a semi-supervised setting to use un-labelled data for data augmentation in separation model [18]. Manuel et al. [11] stated that the quality of training data is key to better generalization of separation models, and proposed the LibriMix dataset for speech separation. With a wider vocabulary, a higher number of distinct speakers, and varying recording conditions, models trained on the LibriMix dataset show better generalization than their counterparts, e.g., WHAM [9] and VCTK [19]. Berkan et al. [12] employed an over-parameterized network [20], with a deeper encoder-decoder architecture, to improve generalization. However, this improvement is modest because the new model doubles the parameters over the original one.

In this paper, we address the generalization issue of separation models in noisy environments. We propose variations of Mixup [21], [22] based augmentation to generate new samples from a linear combination of samples during mini-batch training. Mixup was previously used for classification [21], [22], and here we extend it to speech separation, which is a regression problem. We propose four variations of Mixup and compare their results with a model trained on unaugmented data and models trained on various configurations of SpecAugment [16]. Experimental results suggest an improved generalization by the proposed method on cross-corpus eval-

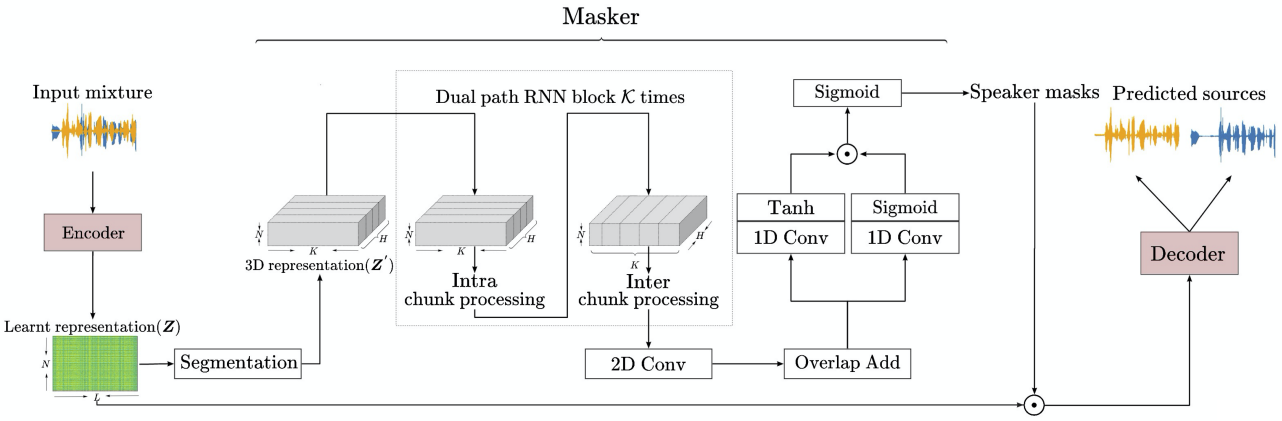


Fig. 1: The architecture of the DPRNN separation model [6]. The model takes an input mixture segment and passes it through an encoder to produce a learnt representation $\boldsymbol{Z}$. $\boldsymbol{Z}$ is segmented into $H$ equal sized chunks of dimension $\mathbb{R}^{N\times K}$ which are concatenated to form a $3D$ tensor $\boldsymbol{Z}^{'}$ which is passed through a series of $\mathcal{K}$ Dual path RNN blocks to perform intra and inter chunk processing for local and global modelling of chunks, respectively. Next, a 2D convolution operation and an overlap add are performed to convert the $3D$ representation back to $2D$ representation. Finally, the output of $1D$ Convolutional gating operation is passed through a Sigmoid activation function to produce masks for each speaker in the mixture. These masks are then multiplied with $\boldsymbol{Z}$ and passed through the decoder to produce the predicted speech sources.

uation with three datasets: LibriMix [11], TIMIT [23] and VCTK [19].

## II. Augmentation for separation

Let $x(t)$ be a single-channel mixture of the clean speeches of $C$ speakers, $\{y_1(t), ..., y_C(t)\}$, and environmental noise $n(t)$:

$$x(t) = \sum_{c=1}^{C} y_c(t) + n(t). \tag{1}$$

We aim to train a separation model $\mathcal{F}(\cdot)$ to retrieve from the mixture the individual speech signals, $\{\hat{y}_1(t), \cdots, \hat{y}_C(t)\}$. To this end, we discuss the separation model and the selection of a data augmentation method to improve the generalizability of the separation model in noisy environments.

### A. Separation model

We select the Dual Path Recurrent Neural Network (DPRNN) [6] model which is a state-of-the-art model for speech separation to evaluate the augmentation strategies. This time-domain model processes the raw waveform in an end-to-end fashion, which helps to reduce the input/output overhead during training. The DPRNN model is based on an encoder $\mathcal{E}(\cdot)$, a masker $\mathcal{M}(\cdot)$ and a decoder $\mathcal{D}(\cdot)$ structure, as shown in Fig. 1.

The model processes the input signal $x(t)$ in short segments. Let a time-domain segment be

$$\bar{\boldsymbol{x}} = [x(1), \cdots, x(W)]^{\mathrm{T}}, \tag{2}$$

where $W$ is the length of the segment and $(\cdot)^{\mathrm{T}}$ represents the transpose.

The encoder $\mathcal{E}(\cdot)$ maps $\bar{\boldsymbol{x}} \in \mathbb{R}^{W\times 1}$ to a learnt latent feature representation $\boldsymbol{Z} \in \mathbb{R}^{N\times L}$ using 1D convolution operation as

$$\mathcal{E}(\bar{\boldsymbol{x}}) = \boldsymbol{Z}, \tag{3}$$

where $N$ is the feature dimension and $L$ the length of each of those feature dimensions.

The masker network first performs a segmentation process which splits $\boldsymbol{Z}$ into $H$ overlapping chunks with a chunk size of length $K$ and hop size $P$. These chunks are concatenated to form a 3D representation $\boldsymbol{Z}^{'} \in \mathbb{R}^{N\times K\times H}$ where $K$ denotes the chunk size and $H$ the number of chunks. The 3D representation is fed to a series of $\mathcal{K}$ dual path bi-directional LSTM layers. Each dual-path LSTM layer performs intra and inter chunk processing of $\boldsymbol{Z}^{'}$ to perform local and global modelling, respectively. The output of the last layer Dual path processing block is passed through a 2D convolutional layer followed by overlap-add to convert the 3D output back to 2D representation and finally a gating operation is applied using 1D convolution layers and output of this gating operation is passed through a Sigmoid layer to predict mask for each source in the mixture. The masker network can be represented as:

$$\mathcal{M}(\boldsymbol{Z}) = \{\boldsymbol{M}_1, \cdots, \boldsymbol{M}_c \cdots, \boldsymbol{M}_C\}, \tag{4}$$

where $\mathcal{M}$ is the masker network and $\boldsymbol{M_c}$ is the mask for the $c_{th}$ source in the mixture. The mask for each source $\boldsymbol{M_c}$ consists of values in the range $[0, 1]$, representing how dominant each element in $\boldsymbol{Z}$ is corresponding to the $c$-th speaker in the mixture.

Following the mask estimation, a representation for each source is computed as:

$$\boldsymbol{D}_c = \boldsymbol{Z} \odot \boldsymbol{M}_c, \quad c = 1, \cdots, C, \tag{5}$$

where $\odot$ represents the element-wise multiplication. Finally, the 1D transpose convolution in the decoder $\mathcal{D}(\cdot)$ maps speaker representation $\{\boldsymbol{D}_1, \cdots, \boldsymbol{D}_C\}$ into a set of predictions, i.e.

$$\bar{\boldsymbol{Y}} = \mathcal{D}(\{\boldsymbol{D}_1, \cdots, \boldsymbol{D}_C\}) = \{\bar{\boldsymbol{y}}_1, \cdots, \bar{\boldsymbol{y}}_{\boldsymbol{c}}, \cdots, \bar{\boldsymbol{y}}_C\}, \tag{6}$$

where $\bar{\boldsymbol{y}}_c = [y_c(1), \cdots, y_c(W)]^{\mathrm{T}}$.

The separation model is trained in a mini-batch style. Each mini-batch contains $B$ segments of speech mixture, i.e.

$$\boldsymbol{X} = [\bar{\boldsymbol{x}}_1, \cdots, \bar{\boldsymbol{x}}_{\boldsymbol{b}}, \cdots, \bar{\boldsymbol{x}}_B], \tag{7}$$

where $\bar{\boldsymbol{x}}_b = [x_b(1), \cdots, x_b(W)]^{\mathrm{T}}$. The corresponding ground truth $\boldsymbol{Y}$ is represented as

$$\boldsymbol{Y} = [\bar{\boldsymbol{Y}}_1, \cdots, \bar{\boldsymbol{Y}}_b, \cdots \bar{\boldsymbol{Y}}_B], \tag{8}$$

where $\bar{\boldsymbol{Y}}_b = [\boldsymbol{y}_{b_1} \cdots, \boldsymbol{y}_{b_C}]$ with $\boldsymbol{y}_{b_c} = [y_{b_c}(1) \cdots y_{b_c}(W)]^{\mathrm{T}}$.

The separation network outputs a mini-batch of predicted waveforms $\hat{\boldsymbol{Y}}$, which can be represented similarly as Eq. (8). The model is trained to minimize the loss between the ground-truth $\boldsymbol{Y}$ and the prediction $\hat{\boldsymbol{Y}}$, the loss function is defined as

$$\mathcal{L}_{\text{regular}} = \mathcal{L}(\boldsymbol{Y}, \hat{\boldsymbol{Y}}) = \sum_{b=1}^{B} \sum_{c=1}^{C} \text{SI-SNR}(\boldsymbol{y}_{\boldsymbol{b_c}}, \hat{\boldsymbol{y}}_{\boldsymbol{b_c}}), \tag{9}$$

where, for a ground-truth $\boldsymbol{y}$ and prediction $\hat{\boldsymbol{y}}$, the scale-invariant signal-noise ratio (SI-SNR) measure is defined as [5], [24]

$$\text{SI-SNR}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = 10 \log_{10} \frac{\|\tilde{\boldsymbol{y}}\|^2}{\|\hat{\boldsymbol{y}} - \tilde{\boldsymbol{y}}\|^2}, \tag{10}$$

where $\tilde{\boldsymbol{y}} = \frac{\langle \hat{\boldsymbol{y}}, \boldsymbol{y} \rangle \boldsymbol{y}}{\|\boldsymbol{y}\|^2}$ and $\langle \hat{\boldsymbol{y}}, \boldsymbol{y} \rangle$ denotes the inner product.

### *B. Augmentation*

Mixup enhances the available training distribution by creating augmented examples from the training mini-batch. Given an input mini-batch $\mathbf{X}$ and $\mathbf{Y}$, the augmented mixture $\mathbf{X}^*$ and ground-truth $\mathbf{Y}^*$ can be generated as

$$\begin{cases} \boldsymbol{X}^* = [\boldsymbol{x}_1^*, \cdots, \boldsymbol{x}_B^*] \\ \boldsymbol{Y}^* = [\boldsymbol{Y}_1^*, \cdots, \boldsymbol{Y}_B^*] \end{cases}, \tag{11}$$

where the $b^{th}$ term is

$$\begin{cases} \boldsymbol{x}_b^* = \lambda \boldsymbol{x}_{i_b} + (1-\lambda) \boldsymbol{x}_{j_b} \\ \boldsymbol{Y}_b^* = \lambda \boldsymbol{Y}_{i_b} + (1-\lambda) \boldsymbol{Y}_{j_b} \end{cases}. \tag{12}$$

Here $i_b$ and $j_b$ are randomly sampled indices from $[1, B]$ and are used to generate the $b$-th segment in the new mini-batch. The scalar $\lambda$ controls the weights between the two components. The value of $\lambda$ is discussed in Sec. III. Mixup based augmentation at batch level has been visualized in Fig. 2. For the ground truth $\boldsymbol{Y}^*$ and the prediction $\hat{\boldsymbol{Y}}^*$, the new loss function is defined

$$\mathcal{L}_{\text{augment}} = \mathcal{L}(\boldsymbol{Y}^*, \hat{\boldsymbol{Y}}^*). \tag{13}$$

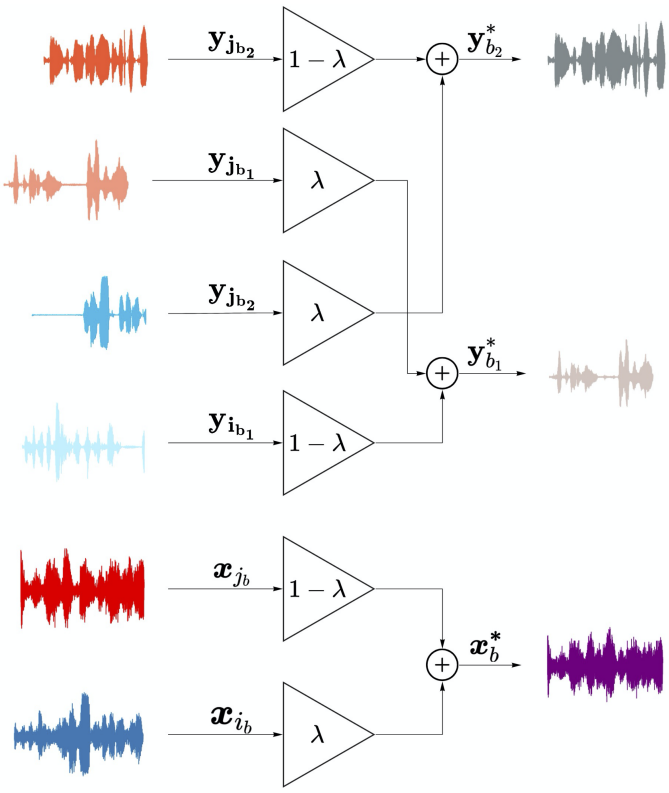


Fig. 2: Mixup data augmentation: two distinct mixtures $\boldsymbol{x}_{i_b}, \boldsymbol{x}_{j_b}$ and their ground-truth speech $\boldsymbol{y}_{i_{b_1}}, \boldsymbol{y}_{i_{b_2}}$ and $\boldsymbol{y}_{j_{b_1}}, \boldsymbol{y}_{j_{b_2}}$ are mixed to produce the new mixtures $\boldsymbol{x}_b^*$ and ground truth $\boldsymbol{y}_{b_1}^*$ and $\boldsymbol{y}_{b_2}^*$.

We proposed four variations of the Mixup methods: Complete, Partial, Pre-trained and Data-only Mixup. Let $e$ be the epoch number going through the training data.

Complete Mixup (CP) uses augmented training for all epochs. The loss function in each mini-batch is defined as

$$\mathcal{L}_{\text{CP}} = \mathcal{L}_{\text{augment}}, \quad 0 < e < E_{\text{max}}, \tag{14}$$

where $E_{\text{max}}$ is the maximum number of epochs we used during training.

Partial Mixup (PA) uses regular training in the first $E_{\text{early}}$ epochs and augmented training every $Q$ epochs afterwards. Initial regular training is done to take advantage of initial convergence speed when the model is not exposed to augmented samples as depicted in the validation loss plot in Fig 4. The loss function is defined as

$$\mathcal{L}_{\text{PA}} = \begin{cases} \mathcal{L}_{\text{regular}}, & 0 < e \leq E_{\text{early}} \\ \mathcal{L}_{\text{regular}}, & (E_{\text{early}} < e < E_{\text{max}}) \wedge (e | Q \neq 0) \\ \mathcal{L}_{\text{augment}}, & (E_{\text{early}} < e < E_{\text{max}}) \wedge (e | Q = 0) \end{cases}. \tag{15}$$

Pre-trained Mixup (PT) uses a pre-trained model with regular training in the $E_{\text{ptrain}}$ epochs and then fine-tune it with

augmented training in all epochs afterwards. The loss function is defined as

$$\mathcal{L}_{\text{PT}} = \begin{cases} \mathcal{L}_{\text{regular}}, & 0 < e \leq E_{\max} \\ \mathcal{L}_{\text{augment}}, & E_{\max} < e < E_{\text{pt}} \end{cases}. \tag{16}$$

Refining a pre-trained model by training it on augmented data ensures that the model is exposed to a wider data distribution: from the original dataset and alternate distribution from Mixup augmentation.

Data-only Mixup (DO) is similar to the complete Mixup, but using a new Mixup function defined as

$$\begin{cases} \boldsymbol{x}_b^\circ = \lambda \boldsymbol{x}_{i_b} + (1-\lambda)\boldsymbol{x}_{j_b} \\ \boldsymbol{Y}_b^\circ = \boldsymbol{Y}_{i_b} \end{cases}. \tag{17}$$

This is essentially close to adding babble noise in form of mixtures from other samples in the mini-batch. We expect this augmentation to increase the model robustness to low energy noise from other mixtures. The loss function is defined as

$$\mathcal{L}_{\text{DO}} = \mathcal{L}(\boldsymbol{Y}^\circ, \hat{\boldsymbol{Y}}^\circ), \quad 0 < e < E_{\max}. \tag{18}$$

The selection of the hyper parameters $E_{\max}$, $E_{\text{early}}$, $E_{\text{pt}}$ and $Q$ will be discussed in Sec. III-B.

## III. Experiments and Discussion

### A. Experimental setup

We compare the performance of three types of separation models: Unaugmented, SpecAugment augmented, and Mixup augmented. Unaugmented models refer to where the input mixture has not been altered before being passed to the network for training. SpecAugment is inspired from [9], and involves masking out random consecutive bands of the time and frequency bins from audio waveform referred to as time and frequency masking respectively [16]. We use three variations of SpecAugment augmentation: *time masking*, *frequency masking* and *time-frequency masking* (T-F). We use four variations of Mixup Augmentation (see section II-B, Complete Mixup (CP), Partial Mixup (PA), Pre-trained Mixup (PT) and Data-only Mixup (DO). All augmentations are randomly applied to 50% of the mini batches during training. We use the Asteroid framework's [25] implementation of DPRNN [6].

We use three datasets (Librimix [11], TIMIT [23] and VCTK [19]) and consider two types of evaluation: intra-corpus and inter-corpus. The former uses Librimix for both training and testing; while the latter uses Librimix for training and uses

TABLE I: Values of the parameters used in the experiment

| Parameters | Equation | value |
|---|---|---|
| $W$ | (2) | 24000 |
| $B$ | (7) | 8 |
| $C$ | (1) | 2 |
| $\alpha$ | (12) | 8.0 |
| $\beta$ | (12) | 1.0 |
| $E_{\max}$ | (14) | 300 |
| $E_{\text{early}}$ | (15) | 30 |
| $E_{\text{pt}}$ | (16) | 100 |
| $Q$ | (15) | 3 |

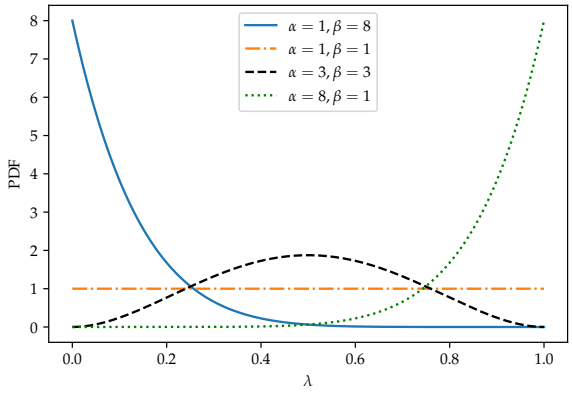


Fig. 3: Probability density function (PDF) of the beta distribution with different $\alpha$ and $\beta$.

TABLE II: SI-SNRi (dB) performance of data-augmented DPRNN for various $(\alpha, \beta)$.

| | | $\beta$ | | |
|---|---|---|---|---|
| | | 1 | 3 | 8 |
| $\alpha$ | 1 | 11.69 | 11.70 | 11.51 |
| | 3 | 11.64 | 11.70 | 11.64 |
| | 8 | **11.97** | 11.70 | 11.57 |

(a) Complete Mixup

| | | $\beta$ | | |
|---|---|---|---|---|
| | | 1 | 3 | 8 |
| $\alpha$ | 1 | 11.17 | 7.60 | 5.89 |
| | 3 | 11.55 | 11.31 | 7.12 |
| | 8 | **12.00** | 11.51 | 11.26 |

(b) Data Only Mixup

TIMIT and VCTK for testing. For training, all the models are trained on train-100-noisy (58hrs) subset of Libri2mix dataset [11]. Libri2mix (train-100-noisy) consists of artificially generated mixtures from the Librispeech corpus with the addition of ambient noise samples from the WHAM [9] test set. The resulting noisy mixtures have a mean SNR -2dB with a standard deviation 3.6dB [11]. We generate Libri2mix (11 hrs), VCTK-2mix (9 hrs) and TIMIT-2mix (10 hrs) for testing. The mixtures in the VCTK test data was created in the same way as the LibriMix noisy samples. The mixture in the TIMIT test data was generated by first randomly mixing utterances from different speakers followed by adding environmental noises from the evaluation set of [26] at each SNRs with an SNR range from -5 to 20dB with a step size of 5. All utterances were sampled at 8 kHz. For this work intra corpus training was restricted to model trained on Librimix as both TIMIT and VCTK test subsets were too small to get a fairly trained separation network.

For performance evaluation, we use the SI-SNR improvement measure [5], which is defined as the difference between the input SI-SNR and output SI-SNR (cf. Eq. (10)) of one segment. Unlike other loss functions, such as signal to distortion ratio (SDR), SI-SNRi is scale invariant and thus suitable for speech applications where proper scaling of speech signal is not ensured [27].

### B. Hyperparameter selection

Table I lists the specific values of the hyperparameters used in the implementation of the paper. The parameter $\lambda \in [0, 1]$

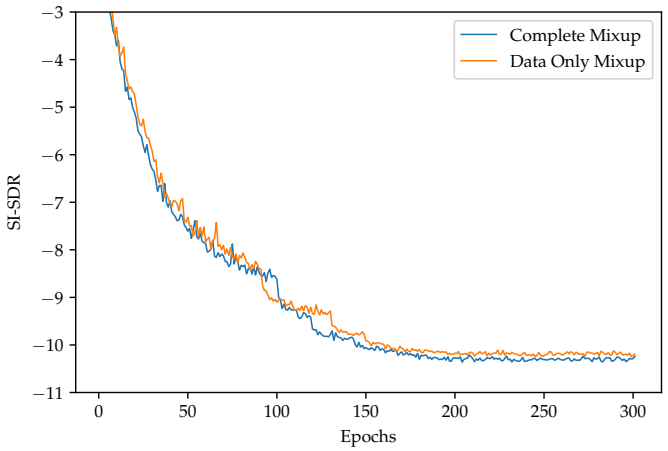


Fig. 4: Validation loss on the validation set of the Librimix dataset when training the DPRNN model with the Complete and Data-only Mixup on the LibriMix dataset.

in Eq. (12)) indicates the amount mixed from $\boldsymbol{x}_{i_b}$ and $\boldsymbol{x}_{j_b}$ to obtain a new mixture $\boldsymbol{x}_b^*$. $\lambda$ is drawn from a beta distribution as $beta(\alpha, \beta)$ [21], where $\alpha$ and $\beta$ determine the shape of the distribution (Fig. 3). We employ a grid search strategy to determine the optimal values of $\alpha$ and $\beta$. During the search, $\alpha$ and $\beta$ both vary from 1 to 8, and the shape of the beta distribution varies from "left skew" to "right skew" correspondingly (Fig. 3). For each ($\alpha$, $\beta$) configuration, we compute the separation performance with two distinct variants of Mixup: Complete and Data-only.

From the results shown in Table II, it can be observed that Complete Mixup is less sensitive to the variation of $\alpha$ and $\beta$ as compared to Data-only Mixup. It can be observed that as the probability of the $\lambda$ value from the $\beta$ distribution gets closer to 1, separation performance is improved as the loss function Eq. (17) is conditioned to optimized the most dominant source in the mixture. Both approaches show the best performance for the configuration ($\alpha = 8, \beta = 1$), corresponding to a right-skewed shape.

All models are trained with a maximum number of epochs $E_{\text{max}}$ unless explicitly stated. $E_{\text{max}}$ is empirically determined by monitoring the tradeoff between validation and training accuracy during training (Fig. 4). Training is stopped if the validation loss does not decrease for 20 consecutive epochs and the model with the best validation loss is selected to prevent overfitting to the training dataset. Adam optimizer [28] with a learning rate of 0.001 is used to train the network. The learning rate is halved if the validation loss does not decrease for 3 consecutive epochs. Gradient clipping with a maximum L2 norm of 5 is used in all experiments. For Partial Mixup, we set $E_{\text{early}}$ as 30 and $Q$ as 3 to first take advantage of initial convergence speed when the model is not exposed to mixed up augmented samples followed by exposure to augmented samples every 3 epochs. Fig. 4 shows examples of convergence curves of DPRNN during training with Complete Mixup augmentation and Data-only augmentation.

TABLE III: Intra-corpus testing results: the performance of various data-augmented DPRNN models trained and tested on Librimix. The average SI-SNRi (dB) across the test set is presented.

| Augmentation type | Augmentation variation | SI-SNRi |
|---|---|---|
| None | - | 12.00 |
| SpecAugment[7] | Frequency masking | 11.63 |
| | Time masking | 12.04 |
| | T-F masking | **12.05** |
| Mixup | Complete | 11.97 |
| | Data-only | 12.00 |
| | Partial | 11.50 |
| | Pre-trained | 12.00 |

TABLE IV: Inter-corpus testing results: the performance of various data-augmented DPRNN models trained on Librimix and tested on TIMIT. The SI-SNRi (dB) for each input SNR [-5, 20] dB is presented. Key - UAUG: Unaugmented, TM: Time masking, FM: Frequency masking, T-F: Time-frequency masking, PA: Partial Mixup, PT: Pre-trained Mixup, CP: Complete Mixup, DO: Data-only Mixup.

| SNR | UAUG | SpecAugment | | | Mixup | | | |
|---|---|---|---|---|---|---|---|---|
| | | TM | FM | T-F | PA | PT | CP | DO |
| -5 | 4.95 | 5.09 | 4.99 | 4.53 | 4.86 | 5.19 | 4.95 | 5.61 |
| 0 | 5.41 | 5.76 | 5.94 | 4.84 | 5.38 | 5.69 | 5.85 | 6.60 |
| 5 | 6.52 | 6.62 | 6.59 | 6.10 | 6.34 | 6.95 | 6.87 | 8.48 |
| 10 | 8.24 | 8.32 | 8.18 | 8.18 | 8.39 | 8.81 | 8.84 | 10.25 |
| 15 | 9.80 | 10.22 | 9.85 | 9.82 | 10.21 | 10.64 | 10.33 | 11.42 |
| 20 | 10.93 | 11.24 | 11.08 | 11.30 | 10.92 | 11.84 | 10.94 | 11.97 |
| Avg | 7.64 | 7.87 | 7.77 | 7.46 | 7.68 | 8.13 | 7.96 | **9.06** |

### C. Results and discussion

The intra-corpus testing results in Table III show that none of the augmentations is able to significantly surpass the unaugmented model in separation performance. Time, Time-Frequency masking; Data-only, Pre-trained and Complete Mixup show comparable performance to the unaugmented model. However, Frequency masking and Complete Mixup perform the worst amongst Specaugmented and Mixup augmented models.

The inter-corpus testing results on TIMIT and VCTK are presented in Table IV and V, respectively. The unaugmented model achieves an SI-SNRi of 12.00dB when trained and tested with Librimix. It can be observed in Table IV and V that the performance of unaugmented models drops significantly in

TABLE V: Inter-corpus testing results: the performance of various data-augmented DPRNN models trained on Librimix and tested on VCTK. The average SI-SNRi (dB) across the test set is presented.

| Augmentation type | Augmentation variation | SI-SNRi |
|---|---|---|
| None | - | 11.07 |
| SpecAugment[7] | Frequency masking | 10.79 |
| | Time masking | 11.09 |
| | T-F | 11.04 |
| Mixup | Complete | 11.11 |
| | Data-only | **11.43** |
| | Partial | 10.93 |
| | Pre-trained | 11.06 |

the case of inter-corpus evaluation. The average performance drop when evaluating on TIMIT dataset (7.64dB SI-SNRi) is more pronounced as compared to on VCTK (11.07dB SI-SNRi). The better performance on VCTK as compared to TIMIT can be attributed to the similar noise samples [9] used to generate mixtures in the test sets of VCTK and Librimix dataset. On the other hand, TIMIT test set has mixtures corrupted with a diverse set of noise samples.

For TIMIT, Data-only Mixup improves the performance by 1.42dB SI-SNRi on average across all SNRs (Table IV). This performance improvement in terms of generalization is significant, especially in noisy environments. However, improvement in separation performance with Partial, Pre-trained and Complete Mixup is marginal. On the other hand, the three variations of SpecAugmented models fail to significantly improve separation performance over the unaugmented model.

Similar to results on TIMIT, Data-only Mixup performs the best on VCTK (Table V). However, the improvement is slight (0.36dB SI-SNRi) compared to improvements on TIMIT (1.42dB SI-SNRi). VCTK and LibriMix mainly differ in the corpus they are derived from [11] and have the same noise types in both their test subset; while the TIMIT test set has mixtures from completely different speech corpus and noise.

From the intra-corpus and inter-corpus testing results, we can summarize that the Data-only Mixup has a desirable trade-off between performance on evaluating on similar and different data distributions compared to the training set. Additionally, the benefit of the Data-only Mixup augmentation seems to increase when noise types differ from the training distribution.

## IV. CONCLUSION

We propose four variations of Mixup based augmentation (Complete, Partial, Pre-trained and Data-only) to improve the cross-corpus generalization of a speech separation model. We compared Mixup augmentation to models trained on SpecAugment and unaugmented data. Experimental results indicate that the proposed method, in particular the Data-only Mixup augmentation, can effectively improve the separation performance when the model is trained and tested on different speech datasets. Unlike approaches that increase the size of the network to cope with noisy conditions for speech separation [12], using a data augmentation approach does not increase the number of parameters of the original network [5]. Future work would include moving from hard-coded to a learned augmentation strategies [29].

## REFERENCES


[1] L. Wang, "Multi-band multi-centroid clustering based permutation alignment for frequency-domain blind speech separation," *Digital Signal Processing*, vol. 31, pp. 79–92, 2014.

[2] M. Kolbæk, D. Yu, Z.-H. Tan, and J. Jensen, "Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 25, no. 10, pp. 1901–1913, 2017.

[3] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *IEEE Intl. Conf. Acoust, Speech Signal Process.*, 2016, pp. 31–35.

[4] Y. Luo and N. Mesgarani, "TasNet: time-domain audio separation network for real-time, single-channel speech separation," in *IEEE Intl. Conf. Acoust, Speech Signal Process.*, 2018.

[5] N. M. Yi Luo, "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.

[6] Y. Luo, Z. Chen, and T. Yoshioka, "Dual-path RNN: efficient long sequence modeling for time-domain single-channel speech separation," in *IEEE Intl. Conf. Acoust, Speech Signal Process.*, 2020.

[7] J. Chen, Q. Mao, and D. Liu, "Dual-path transformer network: Direct context-aware modeling for end-to-end monaural speech separation," in *Proc. Interspeech*, 2020.

[8] E. Nachmani, Y. Adi, and L. Wolf, "Voice separation with an unknown number of multiple speakers," in *Intl. Conf. Machine Learning*, 2020.

[9] Wichern *et al.*, "WHAM!: Extending speech separation to noisy environments," in *Proc. Interspeech*, 2019.

[10] M. Maciejewski, G. Wichern, E. McQuinn, and J. Le Roux, "WHAMR!: Noisy and reverberant single-channel speech separation," in *IEEE Intl. Conf. on Acoust, Speech and Signal Process.*, 2020.

[11] J. Cosentino, M. Pariente, S. Cornell, A. Deleforge, and E. Vincent, "LibriMix: An open-source dataset for generalizable speech separation," *arXiv preprint arXiv:2005.11262*.

[12] B. Kadioglu, M. Horgan, X. Liu, J. Pons, D. Darcy, and V. Kumar, "An empirical study of Conv-TasNet," *IEEE Intl. Conf. Acoust, Speech Signal Process.*, 2020.

[13] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[14] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," *Intl. Conf. for Learning Representations*, 2017.

[15] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Intl. Conf. on machine learning*, 2015.

[16] 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 *Proc. Interspeech*, 2019.

[17] N. Zeghidour and D. Grangier, "Wavesplit: End-to-end speech separation by speaker clustering," *arXiv preprint arXiv:2002.08933*.

[18] M. W. Lam, J. Wang, D. Su, and D. Yu, "Mixup-breakdown: a consistency training method for improving generalization of speech separation models," in *IEEE Intl. Conf. Acoust. Speech Signal Process.*, 2020.

[19] C. Veaux *et al.*, "Superseded-CSTR VCTK corpus: English multi-speaker corpus for cstr voice cloning toolkit," 2016.

[20] B. Neyshabur, Z. Li, S. Bhojanapalli, Y. LeCun, and N. Srebro, "Towards understanding the role of over-parametrization in generalization of neural networks," *Intl. Conf. Learning Representations*, 2019.

[21] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," *Intl. Conf. Learning Representations*, 2018.

[22] 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 *Intl. Conf. Machine Learning*, 2019.

[23] J. S. Garofolo, "TIMIT acoustic phonetic continuous speech corpus," *Linguistic Data Consortium*, 1993.

[24] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "SDR - half-baked or well done?" in *IEEE Intl. Conf. Acoust, Speech Signal Process.*, 2019.

[25] M. Pariente, S. Cornell, J. Cosentino *et al.*, "Asteroid: the PyTorch-based audio source separation toolkit for researchers," in *Proc. Interspeech*, 2020.

[26] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 23, no. 1, pp. 7–19, 2014.

[27] M. Kolbæk, Z.-H. Tan, S. H. Jensen, and J. Jensen, "On loss functions for supervised monaural time-domain speech enhancement," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 825–838, 2020.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Intl. Conf. Learning Representations*, 2014.

[29] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "RandAugment: Practical automated data augmentation with a reduced search space," in *IEEE/CVF Conf. Computer Vision and Pattern Recognition Workshops*, 2020, pp. 702–703.