Speech Communication 152 (2023) 102949

Contents lists available at ScienceDirect

Speech Communication

journal homepage: www.elsevier.com/locate/specom

Data augmentation for speech separation

Ashish Alex[a], Lin Wang[a,*], Paolo Gastaldo[b], Andrea Cavallaro[a]

[a] Centre for Intelligent Sensing, Queen Mary University of London, United Kingdom
[b] Department of Electrical, Electronics and Telecommunication Engineering and Naval Architecture (DITEN), University of Genoa, Italy

ARTICLE INFO

Dataset link: http://www.eecs.qmul.ac.uk/~linwang/demo/augmentation.html

Keywords:
Data augmentation
Deep learning
Domain generalization
Speech separation

ABSTRACT

Deep learning models have advanced the state of the art of monaural speech separation. However, the performance of a separation model considerably decreases when tested on unseen speakers and noisy conditions. Separation models trained with data augmentation generalize better to unseen conditions. In this paper, we conduct a comprehensive survey of data augmentation techniques and apply them to improve the generalization of time-domain speech separation models. The augmentation techniques include seven source-preserving approaches (Gaussian noise, Gain, Time masking, frequency masking, Short noise, Time stretch, and Pitch shift) and three non-source preserving approaches (Dynamix mixing, Mixup, and Cutmix). After hyperparameter search for each augmentation method, we test the generalization of the augmented model by cross-corpus testing on three datasets (LibriMix, TIMIT, and VCTK), and identify the best augmentation combination that enhances generalization. Experimental results indicate that a combination of several non-source preserving strategies (CutMix, Mixup, and Dynamic mixing) resulted in the best generalization performance. Finally, the augmentation combinations also improved the performance of the speech separation model even when fewer training data are available.

1. Introduction

Speech separation is the task of isolating two or more overlapping speech utterances from a mixed speech signal with multiple speakers talking simultaneously. The mixed speech signal can be further corrupted by environmental noise which can make the separation task more difficult. 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 (Wang, 2014; Wang and Cavallaro, 2018), single-channel speech separation is a more challenging task (Wang and Chen, 2018). Deep neural networks (DNN) are at the forefront for speech separation and can be broadly categorized into time–frequency domain approaches (Kolbæk et al., 2017; Hershey et al., 2016; Wang et al., 2018b; Williamson et al., 2015; Wang et al., 2022) and time-domain approaches (Luo and Mesgarani, 2018; Luo and Nima Mesgarani, 2019; Luo et al., 2020; Chen et al., 2020; Nachmani et al., 2020). Time–frequency approaches convert the mixture waveform into the time–frequency domain using the short-time Fourier transform (STFT), separate time–frequency features for each source, then reconstruct the source waveforms by inverse STFT. They usually use the original phase of the mixture to synthesize the estimated source waveforms, which retain the phase of the noisy mixture (Kolbæk et al., 2017; Hershey et al., 2016). This strategy imposes an upper limit on the separation performance. Many methods have been proposed to retrieve the clean phase (Wang et al., 2018b), or are based on the use of complex spectrograms in order to solve this issue (Williamson et al., 2015; Wang et al., 2022). Time-domain approaches perform end-to-end separation by directly modeling the mixture waveform with an encoder–decoder framework, and have made great progress and attracted significant attention in recent years (Luo and Nima Mesgarani, 2019; Luo et al., 2020).

DNN approaches (Luo and Nima Mesgarani, 2019; Luo et al., 2020; Chen et al., 2020) have shown remarkable improvement compared to traditional signal processing methods such as computational auditory scene analysis (Brown and Wang, 2005) and non-negative matrix factorization (Schmidt and Olsson, 2006) in separating mixed speech from the benchmark speech separation WSJ0-2mix(clean) dataset (Hershey et al., 2016). However, the performance of separation models drops in real-world conditions with noise (Wichern et al., 2019; Maciejewski et al., 2020) and when tested on speakers and noise types outside the training conditions (Cosentino et al., 2020; Kadioglu et al., 2020). Although performance improvement has been reported when using cascaded models (Maciejewski et al., 2020; Liu et al., 2020) (separate

[*] Corresponding author.
E-mail addresses: a.alex@qmul.ac.uk (A. Alex), lin.wang@qmul.ac.uk (L. Wang), paolo.gastaldo@unige.it (P. Gastaldo), a.cavallaro@qmul.ac.uk (A. Cavallaro).

https://doi.org/10.1016/j.specom.2023.05.009
Received 2 March 2022; Received in revised form 15 April 2023; Accepted 25 May 2023
Available online 1 June 2023

model for separation and enhancement), this improvement is modest and requires almost doubling the number of network parameters.

A model has good generalization if similar performance is obtained when tested on data outside training distribution (Kadioglu et al., 2020; Hendrycks et al., 2020). Lack of generalization typically stems from overfitting the model on the training dataset. Generalization of models can also be improved with regularization techniques such as dropout (Srivastava et al., 2014), early stopping, weight decay and batch normalization (Zhang et al., 2017; Ioffe and Szegedy, 2015). However, models trained with one or more of these regularization strategies were found to underperform with new test subsets outside their training conditions (Kadioglu et al., 2020). Increasing the representational capacity of the model by adding the number of layers or using a wider network could improve the generalization of models (Zagoruyko and Komodakis, 2016). Other techniques to improve generalization include model optimization (Pariente et al., 2020b), transfer and meta learning (Wu et al., 2021). However, engineering changes to the model, such as finding a new architecture (e.g. Dual-path network (Luo et al., 2020)) is harder than refining the training data itself (Wei et al., 2020).

Data augmentation has been extensively used to improve the generalization (Ko et al., 2015; Wei et al., 2018) and to prevent the overfitting (Hendrycks et al., 2020; Wei et al., 2020). Performance improvement can be obtained when a model is trained with multiple augmentations with optimal hyperparameters (Cubuk et al., 2019; Lim et al., 2019; Zhang et al., 2020; Cubuk et al., 2020). AutoAugment (Cubuk et al., 2019) and its extensions (Lim et al., 2019; Zhang et al., 2020; Cubuk et al., 2020; Ho et al., 2019) primarily focused on training a model that learns a combination of augmentations and its hyperparameters. However, training such a model (e.g. AutoAugment (Cubuk et al., 2019)) requires extensive computational resources as opposed to using a predefined set of augmentations.

In this paper, we conduct a comprehensive survey of data augmentation strategies and empirically evaluate the ability of ten methods to improve the generalization performance of time-domain separation models. The contribution and novelty of the paper is summarized below.

- First, we apply variants of `Mixup` (Alex et al., 2021) and `CutMix` (Yun et al., 2019), which were originally proposed in the computer vision domain, to the speech separation problem, and achieve the top two performance among all the individual augmentation methods.
- Second, we conduct an ablation study to identify the best hyperparameters for each individual augmentation method.
- Third, we improve the generalization performance by empirically searching for the best strategy for combining various augmentation methods. We identify that the combination of `Mixup`, `CutMix` and `Dynamic mixing` augmentation gives the best generalization result.

We apply the augmentation strategies to the training of two popular time-domain speech separation models: DPRRN (Luo et al., 2020) and ConvTasNet (Luo and Nima Mesgarani, 2019), and evaluate the performance of the augmented speech separation model via intra-corpus and cross-corpus testing on three speech datasets (LibriMix (Cosentino et al., 2020), TIMIT (Garofolo, 1993) and VCTK (Veaux et al., 2016)).

The paper is organized as follows: We discuss related works on data augmentation in Section 2, and formulate the problem in Section 3. We adapt the various augmentation methods to speech separation in Section 4, and evaluate the generalization performance in Section 5. Finally, conclusions are presented in Section 6.

2. Related work

2.1. Speech separation in noisy environments

In recent years, the introduction of WHAM! (Wichern et al., 2019) and LibriMix (Cosentino et al., 2020) datasets has accelerated the research of speech separation in noisy environments (Gao et al., 2017). WHAM! dataset is an extension of WSJ0-2mix (clean) dataset with ambient noise samples from bars, restaurants and coffee shops. However, Cosentino et al. (2020) highlighted the performance drop when models trained on WSJ0-2mix and WHAM! were evaluated on other datasets, such as their proposed LibriMix dataset. LibriMix dataset was reported to have lower generalization error than models trained on WHAM! and WSJ0-2mix dataset (Cosentino et al., 2020). This improvement was attributed to the larger size of the dataset, variability in recording conditions and the presence of a higher and diverse range of unique speakers in the LibriMix corpus. Despite the improvement in generalization when using LibriMix datasets (Cosentino et al., 2020) for training there is still a lack of generalization when evaluated outside its test corpus, especially on unseen noisy conditions. Additionally, WHAM! (Wichern et al., 2019), VCTK (Veaux et al., 2016) and LibriMix (Cosentino et al., 2020) all use the same noise corpus (WHAM (Wichern et al., 2019)) in their test subset and thus is not a thorough test of generalization.

A summary of the deep learning models tested on WHAM! and LibriMix dataset has been presented in Table 1. Fig. 1 presents the tradeoff between the performance and number of parameters for speech separation models in clean (WSJO-2mix (Hershey et al., 2016)) and noisy (WHAM (Wichern et al., 2019)) conditions. For both clean and noisy conditions, time domain models largely tend to outperform frequency domain models. Wavesplit (Zeghidour and Grangier, July 2021) was reported to have the best separation performance in both clean and noisy environments on WHAM and LibriMix datasets. However, Wavesplit uses speaker-ids as additional information during training and also has a significantly high number of parameters (29M) as compared to other time-domain models (Luo and Nima Mesgarani, 2019; Luo et al., 2020; Chen et al., 2020). On the other hand, DPTNet (Chen et al., 2020) has the best performance vs model parameters tradeoff. However, DPTNet has very high training time compared to other state-of-the-art time domain separation models e.g. DPRNN (Luo et al., 2020) which makes it impractical for research works with extensive ablation experiments.

Furthermore, from Table 1 we can infer that using cascaded variants of TasNet and Deep CASA models results in only 0.6 and 1 dB SI-SNRi performance improvement with doubling the number of parameters which is further highlighted in Fig. 1 with an "L" shape.

2.2. Augmentations

Data augmentations have been extensively used in varying machine learning domains (e.g. vision (Shorten and Khoshgoftaar, 2019)/audio (Wei et al., 2020)). Data augmentation can encode additional priors other than the once introduced by the choice of model architecture by altering/enhancing the available training data which can enhance the robustness of a model to unseen conditions.

Data augmentation for speech separation can be divided into source-preserving and non-source-preserving augmentations. Most separation augmentation approaches are source preserving in nature, i.e. the augmentation is only applied to the input mixtures and the ground-truth sources are maintained after an augmentation operation is applied (e.g. `SpecAugment` (Park et al., 2019)). Non-source preserving augmentation modifies both the input mixture and its ground truth sources with the augmentation. An example of non-source preserving augmentation is `Mixup` (Zhang et al., 2018a) which linearly mixes both input and its ground truth. Table 2 provides a summary of various augmentations.

Table 1
Summary of speech separation models tested for speech separation in noisy environments on WHAM (8 KHz) (Wichern et al., 2019) and LibriMix (8 KHz noisy) (Cosentino et al., 2020) datasets.

Ref	Algorithm	Domain		Params[M]	SI-SNRi	
		FD	TD		WHAM	LibriMix
Liu et al. (2020)	Chimera++ (Wang et al., 2018a)	✓		29.6	9.9	–
Liu et al. (2020)	Chimera++ casc. (Wang et al., 2018a)	✓		59.2	10.3	–
Wu et al. (2020)	A2PIT (Luo and Mesgarani, 2020)		✓	–	10.9	–
Wu et al. (2020)	SADDEL (Wu et al., 2020)		✓	–	12.0	–
Maciejewski et al. (2020)	TasNet (Luo and Mesgarani, 2018)		✓	32.0	12.0	–
Maciejewski et al. (2020)	TasNet casc. (Luo and Mesgarani, 2018)		✓	64.0	12.6	–
Maciejewski et al. (2020)	ConvTasNet (Luo and Nima Mesgarani, 2019)		✓	5.1	12.7	11.7
Pariente et al. (2020b)	Filterbank (Pariente et al., 2020b)		✓	5.1	12.9	–
Liu et al. (2020)	Deep CASA (Liu and Wang, 2019)	✓		12.8	13.4	–
Liu et al. (2020)	Deep CASA casc. (Liu et al., 2020)	✓		25.6	14.4	–
Nachmani et al. (2020)	DPRNN[a] (Luo et al., 2020)		✓	3.7	13.9	12.0
Nachmani et al. (2020)	DPRNN gated (Nachmani et al., 2020)		✓	7.5	15.2	–
Zeghidour and Grangier (July 2021)	Wavesplit[b] (Zeghidour and Grangier, July 2021)		✓	29.0	15.4	15.1
Zeghidour and Grangier (July 2021)	Wavesplit[b] + DM (Zeghidour and Grangier, July 2021)		✓	29.0	16.0	15.2
Chen et al. (2020)	DPTNet (Chen et al., 2020)		✓	2.7	–	–
Subakan et al. (2021)	SepFormer (Subakan et al., 2021)		✓	26.0	–	–

KEY: FD — frequency domain, TD — time domain, SI-SNRi — Scale invariant Signal to distortion ratio improvement, casc. — cascaded. Params — number of parameters in million(M) in the network. Fields marked as '–' are the ones for which data was not found/published.

[a] Asteroid (Pariente et al., 2020a) implementation.

[b] Uses speaker-ids as additional information.

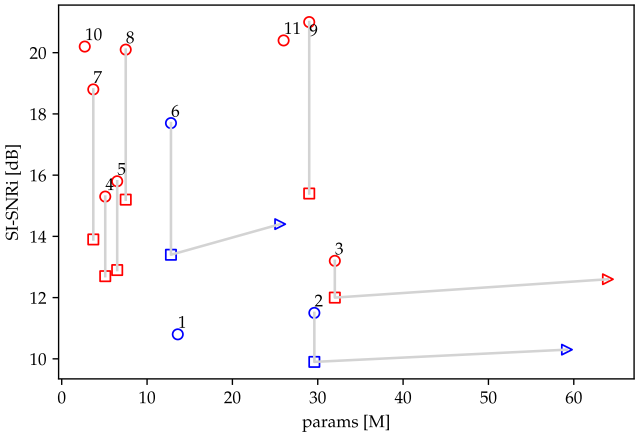


Fig. 1. Relationship between the number of parameters (in million (M)) and performance of model (SI-SNRi) in clean (WSJ0-2mix (Hershey et al., 2016)) and noisy (WHAM! (Wichern et al., 2019)) conditions. Blue and red markers represent frequency and time domain models, respectively. Clean: ○ ○; Noisy: □ □; Cascaded network: ▷ ▷. Models: 1. DPCL 2. Chimera++ 3. TasNet 4. ConvTasNet 5. Filterbank 6. Deep CASA 7. DPRNN 8. DPRNN-gated 9. Wavesplit 10. DPTNet 11. SepFormer. Note: Artificial space has been created between ConvTasNet (Luo and Nima Mesgarani, 2019) and Filterbank (Pariente et al., 2020b) for ease of viewing.

A mini-survey of data augmentation techniques for audio (Wei et al., 2020) reported source preserving augmentations to have a very limited impact in improving the audio classification accuracy. Tested source preserving augmentations included: `Gaussian noise`; `Time stretch`; `Pitch shift` and Time and Frequency masking (`SpecAugment` Park et al., 2019) augmentations. Whereas, non-source preserving augmentations: `SpecMix` (Kim et al., 2021); `Mixup` and their proposed `Mixed Frequency Masking` had the most performance improvements; 1.14 and 1.28 mean average precision percentage improvement respectively to the baseline and thus performance not being very substantial.

Some augmentation techniques used in the audio domain have been borrowed from the vision domain. For example, `SpecAugment` (Park et al., 2019) where random bands of time and frequency bins are masked is very similar to `Cutout` (DeVries and Taylor, 2017) and `Random erasing` (Zhong et al., 2020) augmentation. Yun et al. (2019) combined `Cutout` (DeVries and Taylor, 2017) and `Mixup` (Zhang et al., 2018a) augmentation and proposed `CutMix` (Yun et al., 2019) where instead of removing the data points and replacing them with zeros (`Cutout` (DeVries and Taylor, 2017)) or Gaussian noise (Summers and Dinneen, 2019), they are replaced with data points from other training examples. Drawing inspiration from `CutMix`, Kim et al. (2021) proposed `SpecMix` augmentation for audio classification and enhancement where the augmentation was applied to spectral features. They reported both `SpecAugment` and `Mixup` to perform worse than Un-augmented models for the speech enhancement task

Table 2
Various augmentation techniques can be used to train the machine learning model. Checkmark (✓) on SS (Speech separation) column indicates whether the augmentation was found to be applied in previous research for speech separation.

Type	Method	SS	Ref
Source preserving	Gaussian noise	✓	Salamon and Bello (March 2017)
	Time stretch	✗	Salamon and Bello (March 2017)
	Pitch shift	✗	Salamon and Bello (March 2017)
	SpecAugment	✓	Park et al. (2019)
	Cutout	✗	DeVries and Taylor (2017)
	SamplePairing	✗	Inoue (2018)
	GANs	✗	Shrivastava et al. (2016)
	Smart Augmentation	✗	Lemley et al. (2017)
Non-source preserving	Mixup	✓	Zhang et al. (2018a)
	Cutmix	✗	Yun et al. (2019)
	SpecMix	✗	Kim et al. (2021)
	Between-class	✗	Tokozume et al. (2018b)
	Dynamic mixing	✓	Zeghidour and Grangier (July 2021)

while SpecMix slightly improves the enhancement performance. This is interesting as speech enhancement is closely related to the problem of speech separation, which refers to the task of separating the signal of interest from a mixture that can either be corrupted with another speech signal (speech separation), noise (speech enhancement), or both (Wang and Cavallaro, 2020; Mukhutdinov et al., 2023). However, one cannot assume that an augmentation operation that does well for enhancement tasks will do well for separation tasks and vice-versa. For example, mixed sample data augmentation (Harris et al., 2021) approaches such as Mixup which involves adding other speech utterances to the training sample could have a different impact on a model that is being primarily trained to remove noise (speech enhancement) as compared to a separation model whose primary task is to separate speakers from the mixture.

Similar to Mixup, Between-class learning (Tokozume et al., 2018a) originally proposed for sound recognition (Tokozume et al., 2018b) involves mixing two samples belonging to different classes with a random ratio which are then input into a model which is trained to predict the mixing ratio. Additionally, Between class learning was reported to have the ability to constrict the shape of feature distribution which helps in improving the generalization of the model (Tokozume et al., 2018a). Along similar lines in SamplePairing (Inoue, 2018) new training samples are generated by overlaying the target image with another image from the training dataset. However, different from Mixup, SamplePairing uses the label of the target image therefore not mixing up labels (Inoue, 2018). Also, weights by which the target image is mixed with other image is fixed in SamplePairing, unlike Mixup where the weights are randomly drawn from a beta distribution (Zhang et al., 2018a). Similar to the aforementioned works, Smart augmentation (Lemley et al., 2017) takes multiple training samples from the same class as input to a generative model to output new training data which can reduce the validation loss for the model designed for the underlying task. Although this strategy is employable for the classification tasks, it is not feasible for the source separation task which is a regression problem.

3. Problem definition

Let $x(t)$ be a single-channel mixture of the clean speech of $C \geq 2$ speakers, $\{y_1(t), \dots, y_C(t)\}$, and noise $n(t)$, i.e.

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

We aim to train a separation model $\mathcal{F}(\cdot)$ to retrieve from the mixture the individual speech signals, $\{\hat{y}_1(t), \dots, \hat{y}_C(t)\}$.

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

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

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

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

$$\boldsymbol{X} = [\bar{\boldsymbol{x}}_1, \dots, \bar{\boldsymbol{x}}_{\boldsymbol{b}}, \dots, \bar{\boldsymbol{x}}_B], \tag{3}$$

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

$$\boldsymbol{Y} = [\bar{\boldsymbol{Y}}_1, \dots, \bar{\boldsymbol{Y}}_b, \dots, \bar{\boldsymbol{Y}}_B], \tag{4}$$

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

We apply augmentation to the training data to improve the generalization of the model. Data augmentation on a min-batch is shown in Fig. 2(a). The speech mixture and the ground truth post-augmentation can be represented as:

$$\begin{cases} \boldsymbol{X}_{aug} = [\bar{\boldsymbol{x}}_{aug\_1}, \dots, \bar{\boldsymbol{x}}_{aug\_b}, \dots, \bar{\boldsymbol{x}}_{aug\_B}] \\ \boldsymbol{Y}_{aug} = [\bar{\boldsymbol{Y}}_{aug\_1}, \dots, \bar{\boldsymbol{Y}}_{aug\_b}, \dots, \bar{\boldsymbol{Y}}_{aug\_B}] \end{cases}. \tag{5}$$

Using the augmented mixture as input, the separation network generates a mini-batch of predicted waveforms $\hat{\boldsymbol{Y}}$, which can be represented similarly as Eq. (4). The model is trained to minimize the loss between the ground-truth $\boldsymbol{Y}_{\boldsymbol{aug}}$ and the prediction $\hat{\boldsymbol{Y}}$, the loss function is defined as

$$\mathcal{L}(\boldsymbol{Y}_{aug}, \hat{\boldsymbol{Y}}) = \sum_{b=1}^{B} \sum_{c=1}^{C} \text{SI-SNR}(\boldsymbol{y}_{aug\_b_c}, \hat{\boldsymbol{y}}_{b_c}), \tag{6}$$

where, for a ground-truth $\boldsymbol{y}_{aug}$ and prediction $\hat{\boldsymbol{y}}$, the scale-invariant signal-to-noise ratio (SI-SNR) is defined as (Luo and Nima Mesgarani, 2019; Le Roux et al., 2019)

$$\text{SI-SNR}(\boldsymbol{y}_{aug}, \hat{\boldsymbol{y}}) = 10 \log_{10} \frac{\|\tilde{\boldsymbol{y}}\|^2}{\|\hat{\boldsymbol{y}} - \tilde{\boldsymbol{y}}\|^2}, \tag{7}$$

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

The whole procedure is illustrated in Fig. 2(a). We aim to find the best augmentation strategy that improves the generalization of the speech separation model (e.g. DPRNN in Fig. 2(b)). All augmentations are applied with a probability of 0.5 unless stated (see Section 5.1).

4. Augmentations for separation

4.1. Source-preserving augmentations

In this subsection, we present seven traditional source-preserving augmentations: Gaussian noise, Gain, Time and Frequency masking, Short noise, Time stretch, Pitch shift. Fig. 3 depicts example results obtained from these augmentation methods.

4.1.1. Gaussian noise

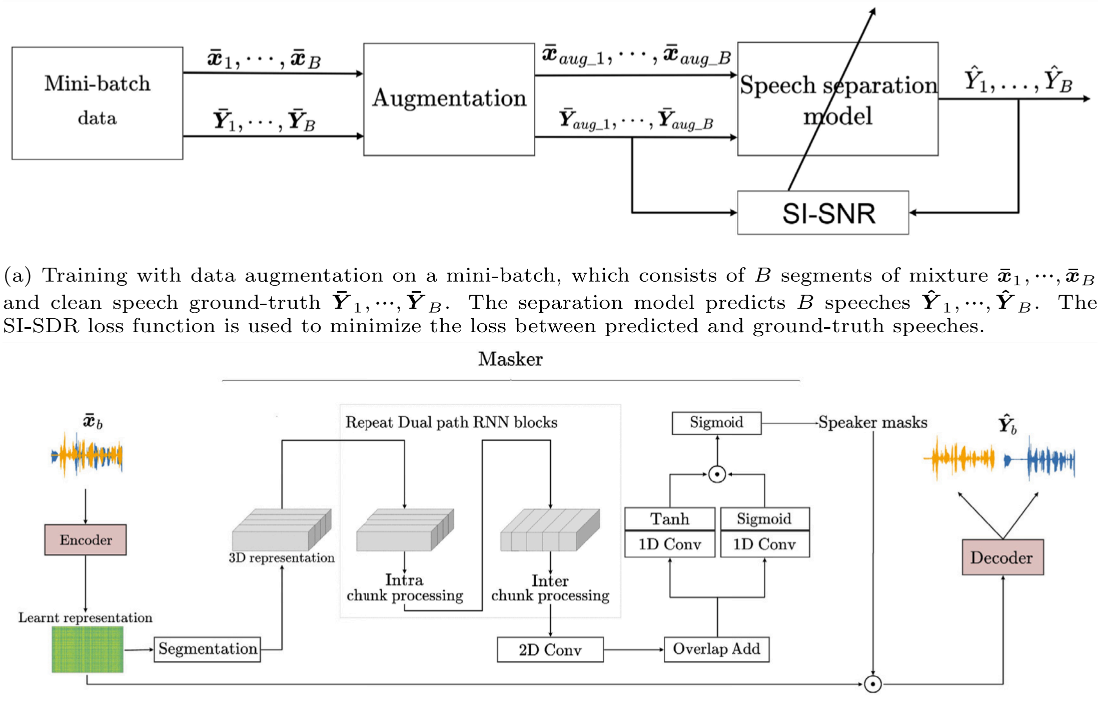
Gaussian noise augmentation adds gaussian noise to the mixture. Adding gaussian noise to the mixture during training can reduce the model's performance sensitivity to mixtures with gaussian noise. Let us use the bth segment in the mini-batch (5) as an example. The augmented data is represented as

$$\begin{cases} \bar{\boldsymbol{x}}_{gs\_b} = \bar{\boldsymbol{x}}_b + A\boldsymbol{g} \\ \bar{\boldsymbol{Y}}_{gs\_b} = \bar{\boldsymbol{Y}}_b \end{cases}, \tag{8}$$

where $\bar{\boldsymbol{x}}_b$ and $\bar{\boldsymbol{x}}_{gs\_b}$ are the original and augmented mixture, respectively; $\boldsymbol{g}$ is the gaussian noise with normal distribution and A is the amplitude, which is randomly sampled from a uniform distribution as

$$A = \mathcal{U}(A_{min}, A_{max}). \tag{9}$$

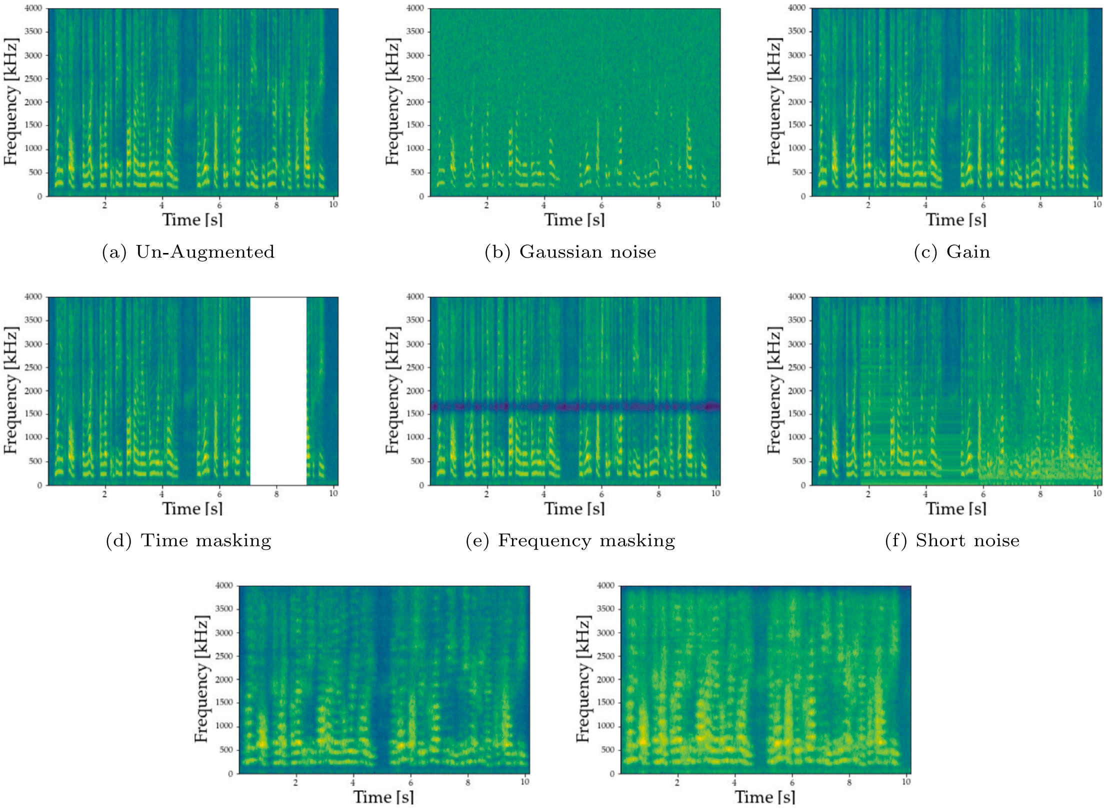
The ablation of the hyperparameters (A_{min}, A_{max}) will be given in Table 4.


Mini-batch data
$\bar{\boldsymbol{x}}_1, \cdots, \bar{\boldsymbol{x}}_B$
$\bar{\boldsymbol{Y}}_1, \cdots, \bar{\boldsymbol{Y}}_B$
Augmentation
$\bar{\boldsymbol{x}}_{aug\_1}, \cdots, \bar{\boldsymbol{x}}_{aug\_B}$
$\bar{\boldsymbol{Y}}_{aug\_1}, \cdots, \bar{\boldsymbol{Y}}_{aug\_B}$
Speech separation model
$\hat{Y}_1, \ldots, \hat{Y}_B$
SI-SNR
Masker
$\bar{\boldsymbol{x}}_b$
Encoder
Learnt representation
Segmentation
3D representation
Repeat Dual path RNN blocks
Intra chunk processing
Inter chunk processing
2D Conv
Overlap Add
Tanh
1D Conv
Sigmoid
1D Conv
Sigmoid
Speaker masks
$\hat{\boldsymbol{Y}}_b$
Decoder

(a) Training with data augmentation on a mini-batch, which consists of B segments of mixture $\bar{\boldsymbol{x}}_1, \cdots, \bar{\boldsymbol{x}}_B$ and clean speech ground-truth $\bar{\boldsymbol{Y}}_1, \cdots, \bar{\boldsymbol{Y}}_B$. The separation model predicts B speeches $\hat{\boldsymbol{Y}}_1, \cdots, \hat{\boldsymbol{Y}}_B$. The SI-SDR loss function is used to minimize the loss between predicted and ground-truth speeches.

(b) DPRNN model for speech separation.

Fig. 2. Data augmentation pipeline for speech separation. We use DPRNN (Luo et al., 2020) as our speech separation model. Details of DPRRN model architecture are presented in Fig. 2(b).



(a) Un-Augmented

(b) Gaussian noise

(c) Gain

(d) Time masking

(e) Frequency masking

(f) Short noise

(g) Time stretch

(h) Pitch shift

Fig. 3. Spectral representation of the seven source-preserving augmentations.

4.1.2. Gain

`Gain` augmentation varies the amplitude of the mixture randomly with a scaling factor. This augmentation aims to increase the robustness of the model to the loudness variation of the mixture. Let us use the bth segment in the mini-batch (5) as an example. The augmented data is represented as

$$\begin{cases} \bar{\boldsymbol{x}}_{gn\_b} = G\bar{\boldsymbol{x}}_b \\ \bar{\boldsymbol{Y}}_{gn\_b} = \bar{\boldsymbol{Y}}_b \end{cases}, \tag{10}$$

where $\bar{\boldsymbol{x}}_b$ and $\bar{\boldsymbol{x}}_{gn\_b}$ are the original and augmented mixture, respectively; G is the scaling factor randomly drawn from an uniform distribution as

$$G = \mathcal{U}(G_{min}, G_{max}). \tag{11}$$

The ablation of the hyperparameters (G_{min}, G_{max}) for gain augmentation can be seen in Table 4.

4.1.3. Time masking

`Time masking` is masking out consecutive time steps from the waveform (Park et al., 2019). `Time masking` augmentation can make the separation model robust in scenarios where a small segment of the audio signal is dropped while recording. Let us use the bth segment in the mini-batch (5) as an example. The augmented data is represented as

$$\begin{cases} \bar{\boldsymbol{x}}_{tm\_b} = \boldsymbol{M}_T \circ \bar{\boldsymbol{x}}_b \\ \bar{\boldsymbol{Y}}_{tm\_b} = \bar{\boldsymbol{Y}}_b \end{cases}, \tag{12}$$

where $\bar{\boldsymbol{x}}_b$ and $\bar{\boldsymbol{x}}_{tm\_b}$ are the original and augmented mixture, respectively; M_T is a binary mask of the same shape as $\bar{\boldsymbol{x}}_b$, and $\circ$ denotes element-wise product.

The W-element vector $\boldsymbol{M}_T$ is defined as

$$\boldsymbol{M}_T(n) = \begin{cases} 0, & \text{if } T_0 \le n \le T_0 + T_1 \\ 1, & \text{otherwise} \end{cases}, \tag{13}$$

where $n \in [0, W-1]$, T_1 is the duration of value-0 segment and is set as

$$T_1 = \mathcal{U}(0, T_{max}W), \tag{14}$$

and T_0 is the start index of the value-0 segment and is set as

$$T_0 = \mathcal{U}(0, W - T). \tag{15}$$

$T_{max} \in [0, 1]$ is the maximum length of the value-0 segment as a fraction of the whole length of the segment. Ablation of the hyperparameter T_{max} has been given in Table 4.

4.1.4. Frequency masking

`Frequency masking` augmentation masks out consecutive frequency bins from the mixture (Park et al., 2019). Using frequency masks can increase the robustness of the separation model when separating mixtures recorded in a cheap microphone, with the speaker at a distance, which can cause the audio signal to have little bass. Also, when the microphone recording the mixture is very close, the low frequency will dominate the spectrum which will be similar to high-frequency components being masked. Let us use the bth segment in the mini-batch (5) as an example. The augmented data is represented as

$$\begin{cases} \bar{\boldsymbol{x}}_{fm\_b} = \mathrm{BS}(\bar{\boldsymbol{x}}_b) \\ \bar{\boldsymbol{Y}}_{fm\_b} = \bar{\boldsymbol{Y}}_b \end{cases}, \tag{16}$$

where $\bar{\boldsymbol{x}}_b$ and $\bar{\boldsymbol{x}}_{tm\_b}$ are the original and augmented mixture, respectively; BS(.) denotes a bandstop filtering.

The bandstop filter is defined as

$$\mathrm{BS}_f = \begin{cases} 0, & \text{if } F_0 \le f \le F_0 + F_1 \\ 1, & \text{otherwise} \end{cases}, \tag{17}$$

where $f \in [0, F_R/2]$ with F_R being the sampling rate F_1 is the width of the stop band and is set as

$$F_1 = \mathcal{U}(0, F_{max}F_R/2), \tag{18}$$

and F_0 is the start frequency of the stop band and is set as

$$F_0 = \mathcal{U}(16, F_R/2 - F_1). \tag{19}$$

The band stop filter is implemented as a Butterworth filter with order 6 (a default value in the Python Scipy library). $F_{max} \in [0, 1]$ is the maximum width of the stop band as a fraction of the whole frequency area. Ablation of the hyperparameter F_{max} will be given in Table 4.

4.1.5. Short noise

The short noise augmentation adds a short burst of noise samples from the noise set of Xu et al. (2015) to the mixture.[1] This can supposedly help with situations where a small burst of noise occurs in between mixtures. Let us use the bth segment in the mini-batch (5) as an example. The augmented data is represented as

$$\begin{cases} \bar{\boldsymbol{x}}_{sn\_b} = \bar{\boldsymbol{x}}_b + \hat{A}\boldsymbol{v} \\ \bar{\boldsymbol{Y}}_{sn\_b} = \bar{\boldsymbol{Y}}_b \end{cases}, \tag{20}$$

where $\bar{\boldsymbol{x}}_b$ and $\bar{\boldsymbol{x}}_{tm\_b}$ are the original and augmented mixture, respectively; $\boldsymbol{v}$ is the additive short noise and $\hat{A}$ is the amplitude.

The additive short noise $\boldsymbol{v}$ is generated as follows. Suppose We have a short noise segment $\bar{\boldsymbol{n}} \in \mathbb{R}^{T_s \times 1}$, we first add fade in and fade out effect to the noise signal with

$$\bar{\boldsymbol{n}}' = \bar{\boldsymbol{n}} \circ \bar{\boldsymbol{G}}, \tag{21}$$

where $\bar{\boldsymbol{G}} \in \mathbb{R}^{T \times 1}$ is a gain vector defined as

$$\bar{\boldsymbol{G}}(n) = \begin{cases} \boldsymbol{F}_{in}, & \text{if } 0 \le n \le T_{in} \\ \boldsymbol{F}_{out}, & \text{if } T_{out} \le n \le T_s \\ 1, & \text{otherwise} \end{cases}, \tag{22}$$

where $\boldsymbol{F}_{in} \in \mathbb{R}^{I \times 1}$ and $\boldsymbol{F}_{out} \in \mathbb{R}^{J \times 1}$ are series of evenly spaced numbers from 0 to 1 and 1 to 0 respectively. The Hyperparameters I and J are sampled from a uniform distribution as:

$$\begin{cases} I = \mathcal{U}(I_{min}, I_{max}) \\ J = \mathcal{U}(J_{min}, J_{max}) \end{cases}. \tag{23}$$

Values of (I_{min}, I_{max}) is set as $(40, 640)$ and (J_{min}, J_{max}) is set as $(80, 800)$ which are the default values in Audiomentations library.[2] We add $\bar{\boldsymbol{n}}'$ to $\bar{\boldsymbol{x}}_b$ at an SNR as

$$\mathrm{SNR}_{sn} = \mathcal{U}(\mathrm{SNR}_{min}, \mathrm{SNR}_{max}). \tag{24}$$

Finally, the faded noise sample $\bar{\boldsymbol{n}}''$ is added to the mixture to obtain short noise-augmented mixture as follows:

$$\bar{\boldsymbol{x}}_{sn_b} = \begin{cases} \bar{\boldsymbol{x}}_b + \bar{\boldsymbol{n}}'' & \text{if } s \le t \le T \\ \bar{\boldsymbol{x}}_b & \text{otherwise} \end{cases}, \tag{25}$$

where s is the start time of the short noise which is randomly sampled from a uniform distribution.

[1] The 104 noise types for training are N1-N17: Crowd noise; N18-N29: Machine noise; N30-N43: Alarm and siren; N44-N46: Traffic and car noise; N47- N55: Animal sound; N56-N69: Water sound; N70-N78: Wind; N79-N82: Bell; N83-N85: Cough; N86: Clap; N87: Snore; N88: Click; N88-N90: Laugh; N91- N92: Yawn; N93: Cry; N94: Shower; N95: Toothbrushing; N96-N97: Footsteps; N98: Door moving; N99-N100: Phone dialing. To compare with the results of [32], N101: AWGN, N102: Babble, N103: Restaurant, N104: Street, were also used.

[2] https://github.com/iver56/audiomentations

4.1.6. Time stretch

Time stretch augmentation speeds up or slows down the mixture without changing the pitch which can make the separation model robust to varying speed perturbations in the audio signal (Arakawa et al., 2019). Let us use the bth segment in the mini-batch (5) as an example. The augmented data is represented as

$$\begin{cases} \bar{\boldsymbol{x}}_{ts\_b} = \mathcal{V}(\bar{\boldsymbol{x}}_b, r_s) \\ \bar{\boldsymbol{Y}}_{ts\_b} = \bar{\boldsymbol{Y}}_b \end{cases}, \tag{26}$$

where $\bar{\boldsymbol{x}}_b$ and $\bar{\boldsymbol{x}}_{tm\_b}$ are the original and augmented mixture, respectively; $\mathcal{V}(\cdot)$ is a phase-vocoder (Laroche and Dolson, 1999) that changes the playing speed of the signal with a ratio r_s, i.e. speeding up for $r_s > 1$ and slowing down for $r_s < 1$. The rate parameter r_s is drawn randomly from a uniform distribution as

$$r_s = \mathcal{U}(r_{min}, r_{max}). \tag{27}$$

It is important to note that we choose the rate parameter such that the mixture is not heavily sped up or slowed down as it would destroy the semantics of the mixture while training the network against the ground truth speakers present in the mixture. Ablation of the hyperparameters (r_{min}, r_{max}) has been given in Table 4.

4.1.7. Pitch shift

Pitch Shift augmentation varies the pitch of the mixture. Varying the pitch of the mixture during training can make the model robust to speakers with their voice having varying phase characteristics (Arakawa et al., 2019). Let us use the bth segment in the mini-batch (5) as an example. The augmented data is represented as

$$\begin{cases} \bar{\boldsymbol{x}}_{ps\_b} = \mathcal{P}(\bar{\boldsymbol{x}}_b, S_p) \\ \bar{\boldsymbol{Y}}_{ps\_b} = \bar{\boldsymbol{Y}}_b \end{cases}, \tag{28}$$

where $\bar{\boldsymbol{x}}_b$ and $\bar{\boldsymbol{x}}_{ps\_b}$ are the original and augmented mixture, respectively; $\mathcal{P}(\cdot)$ is the Pitch shift operation given the semitone parameter S_p.

Pitch shift has a parameter called bins per octave, which is set to 12 as it ensures that 1 step equals one semitone. We shift the waveform by a fixed number of steps S_p, which is randomly drawn from a uniform distribution as

$$S_p = \mathcal{U}(S_{min}, S_{max}), \tag{29}$$

where S_{min} and S_{max} are minimum and maximum semitones.

The Pitch shift operation is conducted as follows. We first compute a rate parameter r_s as:

$$r_s = 2^{-(S_p/12)}. \tag{30}$$

The rate parameter r_s is used to get a time-stretched signal as described in Section 4.1.6, i.e.

$$\bar{\boldsymbol{x}}_{ps\_b} = \mathcal{V}(\bar{\boldsymbol{x}}_b, r_s), \tag{31}$$

Finally, we resample $\bar{\boldsymbol{x}}_{ps}$ with a sampling rate of $S'_r = S_r / r_s$ to obtain the pitch shifted signal.

Ablation of the hyperparameters (S_{min}, S_{max}) will be given in Table 4.

4.2. Non-source preserving augmentations

In this section, we present three non-source preserving augmentation strategies, including one existing method (Dynamix Mixing) and an extension of (Mixup and CutMix) augmentations.

4.2.1. Dynamic mixing

Dynamic mixing augmentation has been used as an augmentation technique in audio speech separation (Subakan et al., 2021). This augmentation strategy attempts to expose the model to a wider range of mixtures. Instead of using fixed training data, Dynamic mixing creates new mixtures from available training data on the fly for each epoch.

We select C unique source segments from the speech corpus $\hat{\boldsymbol{y}}_1, \dots, \hat{\boldsymbol{y}}_C$ to create the augmented data as:

$$\begin{cases} \bar{\boldsymbol{x}}_{dm\_b} = \hat{\boldsymbol{y}}_{b_1} + \cdots + \hat{\boldsymbol{y}}_{b_C} \\ \bar{\boldsymbol{Y}}_{dm\_b} = [\hat{\boldsymbol{y}}_{b_1}, \dots, \hat{\boldsymbol{y}}_{b_C}] \end{cases}. \tag{32}$$

The C sources are selected by sampling from the speech corpus by randomly selecting C distinct indices as:

$$i_c = \mathcal{U}(0, \mathcal{N}) : n = 1, \dots, c, \dots, C, \tag{33}$$

where $\mathcal{N}$ is the total number of mixtures in the speech separation dataset which is 13900 which is the number of mixtures in the train-100 split of LibriMix (Cosentino et al., 2020) dataset. Thus, during each epoch, the model sees new training data instead of having fixed training data for each epoch. Dynamic mixing augmentation is applied with a probability of P for each sample in the mini-batch. Ablation of the impact of P has been presented in Table 4.

4.2.2. Mixup

Mixup enhances the available training distribution by creating augmented examples from the training mini-batch. Mixup is a domain agnostic augmentation technique that can increase the model's robustness to mixtures with babble-like noises as it involves convex combinations with pairs of mixtures and its sources (Zhang et al., 2018a). We propose two variations of Mixup augmentation: Complete Mixup (CP), which generate augmented mixture and ground-truth, and Data-only Mixup (DO), which generates augmented mixture only (Alex et al., 2021).

For Complete Mixup, let us use the bth segment in the mini-batch (5) as an example. The augmented data is represented as

$$\begin{cases} \bar{\boldsymbol{x}}_{cp\_b} = \lambda \bar{\boldsymbol{x}}_{i_b} + (1-\lambda)\bar{\boldsymbol{x}}_{j_b} \\ \bar{\boldsymbol{Y}}_{cp\_b} = \lambda \bar{\boldsymbol{Y}}_{i_b} + (1-\lambda)\bar{\boldsymbol{Y}}_{j_b} \end{cases}, \tag{34}$$

where i_b and j_b are randomly sampled indices from $[1, B]$ and are used to generate the bth augmented segment. The scalar λ is drawn from a beta distribution $beta(\alpha, \beta)$ and controls the weights between the two components. Fig. 4(a) illustrates how Mixup augmentation is applied on a mini-batch. Fig. 4(b) illustrates the relationship between (α, β) and λ.

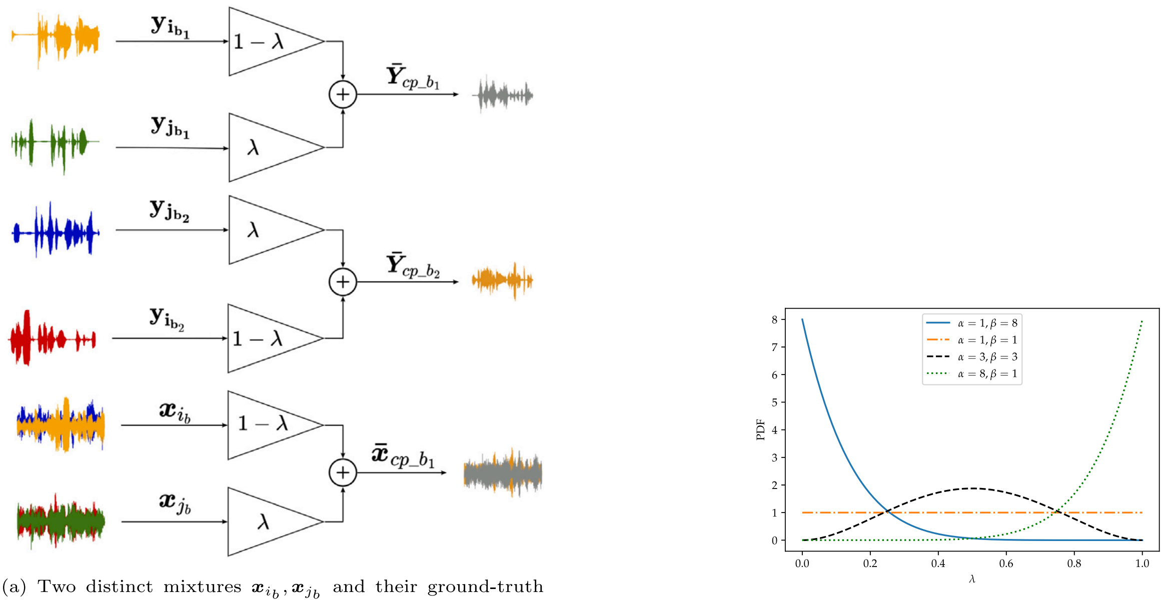
Data-only Mixup (DO) is similar to Complete Mixup, but operates on the mixture only. This is essentially close to adding babble noise in form of mixtures from other samples in the mini-batch. We expect this augmentation to increase the model's robustness in presence of speech-like noises. The augmented data can be represented as

$$\begin{cases} \bar{\boldsymbol{x}}_{do\_b} = \lambda \bar{\boldsymbol{x}}_{i_b} + (1-\lambda)\bar{\boldsymbol{x}}_{j_b} \\ \bar{\boldsymbol{Y}}_{do\_b} = \bar{\boldsymbol{Y}}_{i_b} \end{cases}. \tag{35}$$

Ablation of the hyperparameters (α, β) will be given in Table 4.

4.2.3. CutMix

CutMix augmentation was initially proposed as an augmentation technique for the image domain (Yun et al., 2019). CutMix augmentation is a combination of Cutout and Mixup augmentation and involves masking out certain regions of the image followed by replacing the masked-out portion with a patch from another image from the same mini-batch. CutMix augmentation can make the model robust against corruption in input mixture (Yun et al., 2019).


$\mathbf{y}_{\mathbf{i}_{\mathbf{b}_1}}$
$1-\lambda$
$\bar{\boldsymbol{Y}}_{cp\_b_1}$
$\mathbf{y}_{\mathbf{j}_{\mathbf{b}_1}}$
λ
$\mathbf{y}_{\mathbf{j}_{\mathbf{b}_2}}$
λ
$\bar{\boldsymbol{Y}}_{cp\_b_2}$
$\mathbf{y}_{\mathbf{i}_{\mathbf{b}_2}}$
$1-\lambda$
$\boldsymbol{x}_{i_b}$
$1-\lambda$
$\bar{\boldsymbol{x}}_{cp\_b_1}$
$\boldsymbol{x}_{j_b}$
λ
$\alpha=1, \beta=8$
$\alpha=1, \beta=1$
$\alpha=3, \beta=3$
$\alpha=8, \beta=1$
PDF
0
1
2
3
4
5
6
7
8
0.0
0.2
0.4
0.6
0.8
1.0
λ

(a) Two distinct mixtures $\boldsymbol{x}_{i_b}, \boldsymbol{x}_{j_b}$ and their ground-truth speech $\boldsymbol{y}_{i_{b1}}, \boldsymbol{y}_{i_{b2}}$ and $\boldsymbol{y}_{j_{b1}}, \boldsymbol{y}_{j_{b2}}$ are mixed to produce the new mixtures $\boldsymbol{x}^*_b$ and ground truth $\boldsymbol{y}^*_{b1}$ and $\boldsymbol{y}^*_{b2}$.

(b) Probability density function (PDF) of the beta distribution with different α and β.

Fig. 4. Mixup augmentation.

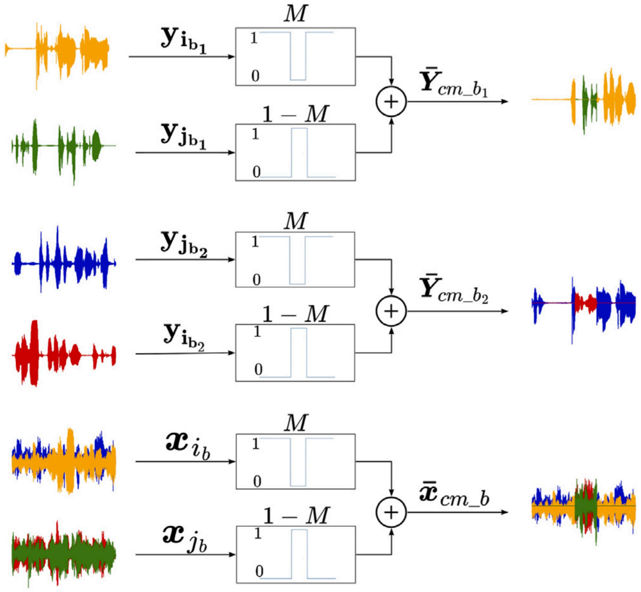

$\mathbf{y}_{\mathbf{i}_{\mathbf{b}_1}}$
M
1
0
$\bar{\boldsymbol{Y}}_{cm\_b_1}$
$\mathbf{y}_{\mathbf{j}_{\mathbf{b}_1}}$
$1-M$
1
0
$\mathbf{y}_{\mathbf{j}_{\mathbf{b}_2}}$
M
1
0
$\bar{\boldsymbol{Y}}_{cm\_b_2}$
$\mathbf{y}_{\mathbf{i}_{\mathbf{b}_2}}$
$1-M$
1
0
$\boldsymbol{x}_{i_b}$
M
1
0
$\bar{\boldsymbol{x}}_{cm\_b}$
$\boldsymbol{x}_{j_b}$
$1-M$
1
0

Fig. 5. Cutmix data augmentation on a mini-batch for 2 speaker mixture: Cutmix combines two distinct mixtures $\boldsymbol{x}_{i_b}, \boldsymbol{x}_{j_b}$ and their ground truth speaker waveforms $\mathbf{y}_{i_{b_1}}, \mathbf{y}_{i_{b_2}}$ and $\mathbf{y}_{j_{b_1}}, \mathbf{y}_{j_{b_2}}$ to produce new mixtures $\bar{\boldsymbol{x}}_{cm\_b}$ and its ground truth speakers $\bar{\boldsymbol{Y}}_{cm\_b_1}$ and $\bar{\boldsymbol{Y}}_{cm\_b_2}$. The rectangular block consists of binary masks M and $1-M$ which undergo an elementwise multiplication operation with the audio signal.

Let us use the bth segment in the mini-batch (5) as an example. The augmented data is represented as

$$\begin{cases} \boldsymbol{x}_{cm\_b} = \boldsymbol{M} \circ \bar{\boldsymbol{x}}_{i_b} + (1-\boldsymbol{M}) \circ \bar{\boldsymbol{x}}_{j_b} \\ \boldsymbol{Y}_{cm\_b} = \boldsymbol{M} \circ \bar{\boldsymbol{Y}}_{i_b} + (1-\boldsymbol{M}) \circ \bar{\boldsymbol{Y}}_{j_b} \end{cases}, \tag{36}$$

where i_b and j_b are randomly sampled indices from $[1, B]$ and are used to generate the bth augmented segment, $\boldsymbol{M}$ is a binary mask of the same length as $\bar{\boldsymbol{x}}_{i_b}$.

The binary mask can be computed as follows. We first calculate a threshold τ which determines the portion of a segment to be mixed, i.e.

$$\tau = \mathcal{U}(0, \tau_{max}), \tag{37}$$

where τ_{max} is a scalar that determines the maximum portion in samples of $\bar{\boldsymbol{x}}_{j_b}$ that will be mixed with $\bar{\boldsymbol{x}}_{i_b}$. Following this we determine start (τ_s) and end (τ_e) point at which segment from $\bar{\boldsymbol{x}}_{j_b}$ will be added to the masked segment of $\bar{\boldsymbol{x}}_{j_b}$ as $\begin{cases} \tau_s = \mathcal{U}(0, W-\tau) \\ \tau_e = \tau_s + \tau \end{cases}$. The binary mask $\boldsymbol{M} \in \{0,1\}^{W \times 1}$ is obtained as

$$\boldsymbol{M}(n) = \begin{cases} 1, \tau_s < n < \tau_e \\ 0, \text{otherwise} \end{cases}. \tag{38}$$

Fig. 5 illustrates the process of CutMix augmentation.

Augmented examples from CutMix tend to be more perceptually perceivable compared to Mixup (Yun et al., 2019). Although, It has been argued that while training a model, the machine's perception takes precedence over that of humans (Summers and Dinneen, 2019). The ablation of the hyperparameters τ_{max} has been given in Table 4.

5. Experiments & discussion

5.1. Experimental setup

We compare various augmentations discussed in Section 4 to the Un-Augmented model. Un-Augmented model refers to where the input mixture has not been altered before passing to the network for training. We start by doing a hyperparameter search for all the augmentations to determine the best-performing hyper-parameter for the particular augmentation. All augmentations are randomly applied to 50% of the mini-batches during training except for Dynamic Mixing where we do ablation with varying probability values as it does not have any other hyper-parameter associated with the augmentation. We train our separation models for 2 speaker separation ($C = 2$) on segments of length 3 seconds with a sample rate $S_r = 8000$ totaling to ($W = 24000$) samples. All models are trained for $E_{max} = 300$ epochs unless specifically stated.

We use three datasets (Librimix (Cosentino et al., 2020), TIMIT (Garofolo, 1993) and VCTK (Veaux et al., 2016)) and consider two types of evaluation: intra-corpus and inter-corpus. The former uses

Table 3
Information about the datasets used in the experiment.

Dataset	Split	Hours	Speakers	Noise corpus
LibriMix (Cosentino et al., 2020)	train-100	58	251	WHAM (Wichern et al., 2019)
LibriMix (Cosentino et al., 2020)	test	11	40	WHAM (Wichern et al., 2019)
VCTK (Yamagishi et al., 2019)	test	9	109	WHAM (Wichern et al., 2019)
TIMIT (Garofolo, 1993)	test	10	630	Env. Noise Corpus (Xu et al., 2015)

Librimix for both training and testing; while the latter uses Librimix for training and uses TIMIT and VCTK for testing. For training, all the models are trained on train-100-noisy (58 h) subset of Libri2mix dataset (Cosentino et al., 2020). Libri2mix (train-100-noisy) consists of artificially generated mixtures from the Librispeech corpus with the addition of ambient noise samples from the WHAM (Wichern et al., 2019) test set. The resulting noisy mixtures have a mean SNR of −2 dB with a standard deviation of 3.6 dB (Cosentino et al., 2020). We generate Libri2mix (11 h), VCTK-2mix (9 h) and TIMIT-2mix (10 h) for testing. The mixtures in the VCTK test data were 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 Xu et al. (2015) at each SNRs with an SNR range from −5 to 20 dB with a step size of 5. All utterances were sampled at 8 kHz, because speech intelligibility mainly requires the information below 4 kHz. For this work, intra-corpus training was restricted to models trained on Librimix as both TIMIT and VCTK test subsets were too small to get a fairly trained separation network. A summary of the three datasets used has been presented in Table 3. It is important to note that the test sets of LibriMix and VCTK datasets are drawn from the same noise corpus (WHAM (Wichern et al., 2019)). Whereas, the test set of the TIMIT dataset is drawn from a different noise corpus (Environmental Noise Corpus (Xu et al., 2015)).

We use the DPRNN (Luo et al., 2020) model, which is a state-of-the-art time-domain speech separation model for all our experiments. We use the Asteroid framework's (Pariente et al., 2020a) implementation of DPRNN (Luo et al., 2020) as shown in Fig. 2(b), DPRNN model has an encoder-masker-decoder architecture. The 1-D convolutional encoder learns a 2-D representation from the given mixture. This 2-D feature representation is divided into 3-D chunks which are then processed by the masker network in a dual-path manner where the masker network performs Intra and Inter chunk processing for local and global modeling of chunks, respectively, to output individual masks for each source in the mixture. Finally, the 1-D transpose convolutional decoder outputs the individual waveforms for each source in the mixture. Dual-path models (Luo et al., 2020; Chen et al., 2020) have been reported to have better separation performance with a lower number of parameters size as compared to the vanilla time-domain models (Luo and Mesgarani, 2018; Luo and Nima Mesgarani, 2019). But it should be noted that this performance improvement is only substantial when the stride and kernel size in the 1-D convolutional encoder is low (Luo et al., 2020). Using lower kernels, stride sizes (e.g. kernel size 2 and stride 1) significantly increases the training time by 1 to 1.5 days on Nvidia Tesla V100 GPU as opposed to using a kernel size 16 and stride 8. Thus we chose a kernel size of 16 and stride 8 for the 1-D convolutional encoder in DPRNN to accommodate for the aforementioned trade-off.

For performance evaluation, we use the SI-SNR improvement measure (Luo and Nima Mesgarani, 2019) as the primary evaluation metric, which is defined as the difference between the input and output SI-SNR (cf. Eq. (7)) of one segment. Unlike other performance measures such as signal-to-distortion ratio (SDR), SI-SNRi is scale-invariant and thus suitable for speech applications where proper scaling of the speech signal is not ensured (Wang and Chen, 2018). Additionally, we evaluate the models with best-performing hyperparameters using perceptual evaluation of speech quality (PESQ) and Short-time objective intelligibility (STOI) which measure the quality and intelligibility of speech respectively. PESQ [−0.5, 4.5] and STOI [0, 1] are widely used in speech enhancement research works and some speech separation research. Both PESQ and STOI have been reported to be closely related to how humans perceive speech (Zhang et al., 2018b) with higher values indicative of better quality and intelligibility respectively.

5.2. Hyperparameter selection

We start by doing a hyperparameter search on each augmentation to identify the best set of hyper-parameters for speech separation. We retain the hyperparameters for the best-performing augmentations for our later experiments based on this hyperparameter search. Additionally, we compare the performance of augmented models with Un-augmented models to get an intuition on augmentations that improve separation performance. Un-Augmented DPRNN model has an SI-SNRi of 12.00 dB on the test set of the LibriMix dataset. It is important to note that in this Section 5.2 all results presented are models trained on (train-100) split of the LibriMix dataset and tested on the test set of LibriMix dataset. Results of this hyperparameter search are presented in Table 4.

`Gaussian Noise`: For ablation of hyperparameters for `Gaussian noise` augmentation we vary the maximum amplitude (A_{max}) with which Gaussian noise is added. The value of A_{max} is progressively decreased from 0.120 to 0.015. Results indicate that increasing A_{max} decreases the separation performance which is expected as adding gaussian noise of higher amplitude severely distorts the resulting augmented mixture. We get the best result from $A_{max} = 0.015$.

`Gain`: Results indicate that varying the scale in which gain is applied does not lead to many variations in separation performance. This is because DPRNN (Luo et al., 2020) model internally uses batch normalization (Ioffe and Szegedy, 2015) which makes the model invariant to the loudness of the mixture.

`Dynamic Mixing`: We test various probabilities with which `Dynamic mixing` can be applied to the mini-batch. Results indicate that increasing the probability of augmentation decreases the separation performance. This is because when the probability of the augmentation is higher, the lower the chances the model sees the same training data throughout training epochs. The empirical results indicate that `Dynamic mixing`, when applied with a probability of 0.25 and 0.5, improves the performance over the Un-Augmented model. The best performance is obtained when `Dynamic mixing` is applied with a probability of 0.5.

`Time masking`: For the ablation of `Time masking`, we vary the maximum amount of time segments that can be masked as a fraction of the total length of the segment which is determined by T_{max} in Eq. (12). Also, when there is a higher chance for a larger amount of input mixture to be masked e.g $T_{max} = 0.40$ the performance of the augmented model starts to deteriorate. In our experiments, we get the best separation performance with $T_{max} = 0.20$.

`Frequency masking`: Similar to `Time masking`, in case of `Frequency masking`, increasing F_{max} decreases the performance of the separation model. Performance degradation with excessive masking is expected as the model will have less of a mixture to predict the sources from. In our experiments, we get the best separation performance with $F_{max} = 0.77$.

`Short noise`: In the case of `Short noise` augmentation, we vary the SNR range in which short noises are added. Results indicate that separation performance deteriorates when short noise is added at very low SNR. This might be happening because of two reasons. Firstly, the test set of the LibriMix dataset has noises from a different noise corpus (WHAM (Wichern et al., 2019)) than the one used for `Short`

Table 4
Ablation of hyperparameters of various data augmentations used to train DPRNN (Luo et al., 2020) model tested on noisy Librimix dataset.

Ref	Augmentation	SI-SNRi	Eq. no.	Hyperparameters	
				A_{min}	A_{max}
Salamon and Bello (March 2017)	Gaussian noise	11.24	(9)	0.001	0.120
		11.58			0.060
		11.87			0.030
		12.00			0.015
				G_{min}	G_{max}
–	Gain	**11.89**	(11)	−6	6
		11.85		−12	12
		11.80		−12	24
				T_{max}	
Park et al. (2019)	Time masking	11.14	(12)	0.60	
		11.45		0.40	
		12.04		0.20	
		12.02		0.10	
				F_{max}	
Park et al. (2019)	Frequency masking	11.09	(16)	0.50	
		11.01		0.25	
		11.77		0.10	
		11.63		0.05	
				P	
Zeghidour and Grangier (July 2021)	Dynamic mixing	12.04	(32)	0.25	
		12.11		0.50	
		11.75		0.75	
		6.75		1.00	
				SNR_{max}	SNR_{min}
–	Short noise	11.49	(24)	−10	0
		11.64		−6	6
		11.34		0	6
		11.70			12
		12.06			24
				S_{min}	S_{max}
Salamon and Bello (March 2017)	Pitch Shift	2.49	(29)	−4	4
		4.26		1	2
				r_{min}	r_{max}
Salamon and Bello (March 2017)	Time stretch	7.96	(27)	0.8	1.25
		9.85		0.9	1.00
				α	β
Alex et al. (2021)	Complete Mixup	11.69	(34)	1	1
		11.70		1	3
		11.51		1	8
		11.64		3	1
		11.70		3	3
		11.64		3	8
		11.97		8	1
		11.70		8	3
		11.57		8	8
				α	β
Alex et al. (2021)	Data-only Mixup	1.17	(35)	1	1
		7.60		1	3
		5.89		1	8
		11.55		3	1
		11.31		3	3
		7.12		3	8
		12.00		8	1
		11.51		8	3
		11.26		8	8
				τ_{max}	
Yun et al. (2019)	CutMix	11.98	(37)	800	
		12.11		2000	
		12.01		4000	
		11.88		8000	
		11.86		12000	

`noise` augmentation (Env noise corpus (Xu et al., 2015)). Secondly, the SNR range of added short noises may not match that of the test set of the LibriMix dataset. The best results on the test set of LibriMix are obtained when using $\text{SNR}_{min} = 0$ dB and $\text{SNR}_{max} = 24$ dB.

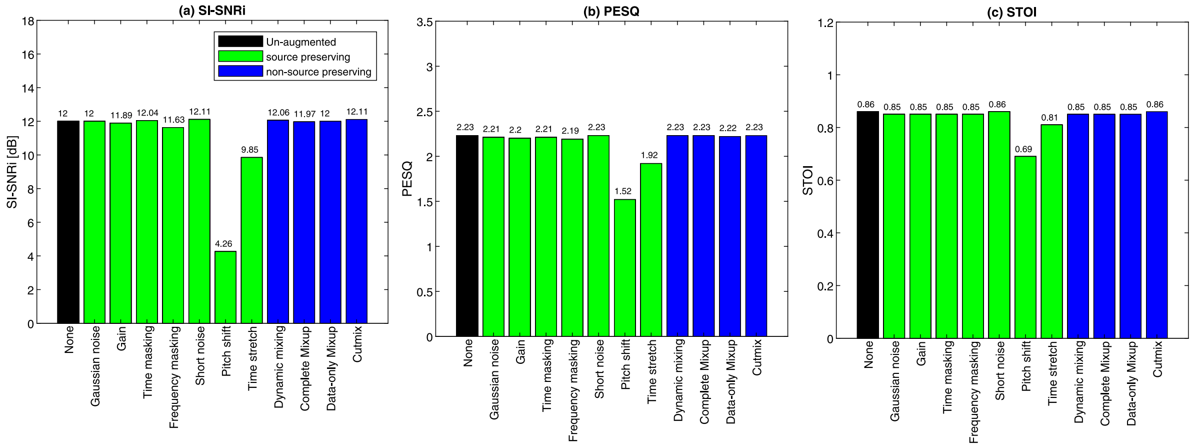


Fig. 6. Results from evaluating various augmentations with best-performing hyperparameters in intra-corpus tests using DPRNN (Luo et al., 2020) model trained and tested on noisy Librimix dataset.

`Pitch shift`: From the ablation of hyper-parameters of `Pitch shift` augmentation we can infer that `Pitch shift` augmentation severely deteriorates the separation performance as compared to the Un-Augmented model. Therefore, for our later experiments, we will forgo this augmentation.

`Time stretch`: Similar to `Pitch Shift` augmentation, `Time stretch` augmentation also severely distorts the separation performance and thus we do not consider this augmentation for our later experiments.

`Mixup`: The hyperparameter $\lambda \in [0,1]$ in Eq. (34) indicates the amount mixed from $\boldsymbol{x}_{i_b}$ and $\boldsymbol{x}_{j_b}$ to obtain a new mixture $\boldsymbol{x}^*_b$. Ablation of hyperparameters α and β for the two distinct variants of `Mixup`: `Complete Mixup` and `Data-only Mixup` from our previous work (Alex et al., 2021) has been presented in Table 4. It can be observed that `Complete Mixup` is less sensitive to the variation of α and β as compared to `Data-only Mixup`. Also, as the probability of the λ value from the β distribution gets closer to 1, separation performance is improved as the loss function Eq. (35) is conditioned to optimize the most dominant source in the mixture. Our results indicated that the best results were achieved with $\alpha = 8$ and $\beta = 1$ for both `Complete` and `Data-only Mixup`.

`CutMix`: Ablation of the maximum number of samples (τ) that will be added from the randomly selected mixture from the mini-batch to the mixture to be augmented. Results indicate that the performance of the `CutMix` augmented models on average are largely invariant to τ and improves the separation performance over the Un-Augmented model. We suppose this is because mixtures used to perform `CutMix` augmentation are from the same mini-batch and thus there is not much variation in the data used for training even when τ value is higher. We get the best results using `CutMix` augmentation with $\tau_{max} = 2000$ (0.25 s).

From our initial hyper-parameter ablation experiments of various augmentation techniques presented in Section 5.2, we present the results from best-performing hyper-parameters for all augmentations when tested on the test set of LibriMix dataset in Fig. 6. None of the augmentations significantly outperform the Un-augmented model. More specifically, `Time stretch` and `Pitch shift` augmentations were the worst-performing augmentations. `Complete` and `Data-only Mixup`, `Time masking`, `Time--Frequency masking`, `Gaussian noise` had comparable performance to the Un-augmented model. On the other hand, `CutMix`, `Dynamic mixing` and `Short noise` showed very minor improvements over the Un-augmented model.

5.3. *Combining augmentations*

We test combining various individual augmentation operations to the combination of `Dynamic mixing` and `CutMix` augmentation. We choose the best-performing hyperparameters (Table 4) for each augmentation to test whether the combination of augmentations leads to improved separation performance.[3] Based on results presented in Table 5 we can concur that separating mixtures from the TIMIT dataset is the hardest as both speech and noise corpus are different than that used in LibriMix dataset. The Un-augmented model has an SI-SNRi of 7.64 dB on the TIMIT dataset compared to 12.00 dB for LibriMix. Additionally, the UnAugmented model on TIMIT has approximately 0.16 lower PESQ than LibiMix and VCTK datasets which is indicative of the lower perceptual quality of separated speech from the test set of TIMIT dataset as compared to the LibriMix dataset. On the other hand, since the VCTK dataset has the same noise corpus as the LibriMix dataset with only the speech corpus being different; separation models performance tested on the VCTK dataset is much closer (10.95 dB SI-SNRi, 2.24 PESQ, 0.77 STOI) to that of the models tested with LibriMix dataset.

We get the best separation performance in terms of SI-SNRi on the TIMIT dataset with `Data-only Mixup` (9.06 dB) which is very closely matched with (`CmixDoDmix`) (9.04 dB). Data-only Mixup particularly has better performance on the TIMIT dataset. However, `Data-only Mixup` only has comparable performance on the LibriMix dataset and only a slight (0.36 dB) SI-SNRi improvement on the VCTK dataset as compared to the Un-Augmented model. On the other hand, we were able to improve upon the separation performance from `Data-only Mixup` on both LibriMix (0.56 dB) and VCTK (0.68 dB) datasets with `Dynamic mixing` + `CutMix` (`CmixDmix`) augmentation combination relative to the Un-Augmented model. Separation performance is further enhanced by adding `Data-only Mixup` to `CmixDmix` (CmixDoDmix). Augmentation combination `CmixDoDmix` has the best average performance across the three tested datasets. Specifically, on the VCTK dataset, `CmixDoDmix` achieves; 0.73 dB SI-SNRi, 0.01 STOI, and 0.06 PESQ improvement over the Un-Augmented model. Whereas, on the TIMIT dataset `CmixDoDmix` has comparatively better inter-corpus performance improvement with 1.40 dB SI-SNRi, 0.02 STOI, and 0.14 PESQ improvement over the Un-Augmented model on TIMIT dataset. Improved performance on the inter-corpus TIMIT and VCTK datasets is indicative of improved generalization of the separation model with varying speech and noise corpus.

[3] Audio samples of separated mixtures with a model trained using various augmentations can be found at http://www.eecs.qmul.ac.uk/~linwang/demo/augmentation.html.

Table 5
Results from combining augmentation techniques on LibriMix, VCTK, and TIMIT datasets.

Augmentation								SI-SNRi[dB]			STOI			PESQ		
DM	CM	DO	CP	GN	SN	TM	FM	Intra-corpus	Inter-corpus		Intra-corpus	Inter-corpus		Intra-corpus	Inter-corpus	
								LibriMix	VCTK	TIMIT	LibriMix	VCTK	TIMIT	LibriMix	VCTK	TIMIT
-	–	–	–	–	–	–	–	12.00	10.95	7.64	0.86	0.77	0.77	2.23	2.24	2.06
-	–	–	–	–	–	–	x	11.63	10.68	7.77	0.85	0.77	0.77	2.19	2.18	2.06
-	–	–	–	–	–	x	–	12.04	10.6	7.47	0.85	0.76	0.76	2.21	2.18	2.05
-	–	–	–	–	x	–	–	12.06	11.23	8.03	0.86	0.77	0.77	2.23	2.26	2.09
x	–	–	–	–	–	–	–	12.11	11.08	7.55	0.85	0.77	0.76	2.23	2.24	2.07
-	x	–	–	–	–	–	–	11.97	10.84	7.76	0.86	0.77	0.77	2.23	2.22	2.06
-	–	–	x	–	–	–	–	12.00	11.02	7.96	0.85	0.77	0.77	2.23	2.24	2.06
-	–	x	–	–	–	–	–	12.00	11.31	**9.06**	0.85	0.78	0.79	2.22	2.26	2.13
-	–	–	–	–	–	x	x	12.05	11.00	7.46	0.85	0.77	0.77	2.22	2.22	0.55
x	x	–	–	–	–	–	–	**12.56**	11.63	8.27	0.86	0.78	0.78	**2.30**	**2.30**	2.17
x	x	–	x	–	–	–	–	12.26	11.34	8.46	0.86	0.77	0.78	2.25	2.25	2.15
x	x	x	–	–	–	–	–	12.55	**11.68**	9.04	0.86	0.78	**0.79**	**2.30**	**2.30**	**2.20**
x	x	–	–	x	–	–	–	12.31	11.20	8.28	0.86	0.77	0.78	2.26	2.24	2.14
x	x	–	–	–	x	–	–	12.29	11.35	7.93	0.86	0.77	0.77	2.27	2.29	2.11
x	x	–	–	–	–	x	–	12.13	10.99	7.71	0.85	0.77	0.77	2.23	2.21	2.08
x	x	–	–	–	–	–	x	11.50	10.78	7.34	0.84	0.77	0.76	2.13	2.15	2.03
x	x	–	–	–	–	x	x	11.80	10.8	7.80	0.85	0.77	0.77	2.17	2.17	2.09
x	x	–	–	–	x	x	–	12.02	10.91	8.36	0.85	0.77	0.77	2.22	2.21	2.12
x	x	–	–	–	x	–	x	12.15	11.34	8.90	0.86	0.78	**0.79**	2.24	2.25	2.18
x	x	–	–	–	x	x	x	11.93	10.87	8.48	0.85	0.77	0.78	2.21	2.19	2.12

KEY: DM — Dynamic mixing, CM — CutMix, DO — Data-only Mixup, CP — Complete Mixup, GN — Gaussian noise, SN — Short noise, TM — Time masking, FM — Frequency masking. Symbol x indicates that the augmentation was used in the combination and – indicates that the augmentation was excluded from the augmentation combination.

From investigating the PESQ and STOI values in Table 5 we can conclude that the minor quantitative improvements for the separation task are hard to interpret. Additionally, Zhang et al. (2018b) reported very minor PESQ and STOI improvements with their separation models despite directly using both metrics as loss functions while training their separation model. Also, most recent research works primarily report SI-SNRi as the primary evaluation metric (Luo et al., 2020; Luo and Nima Mesgarani, 2019; Zeghidour and Grangier, July 2021; Subakan et al., 2021; Michelsanti et al., 2021). To this end, we will be using SI-SNRi as our primary evaluation metric of focus for the rest of the paper.

Among the source preserving augmentations listed in Fig. 6; `Short noise` augmentation showed the best performance improvement on TIMIT dataset (0.39 dB) with comparable performance to the Un-Augmented model on LibriMix and VCTK datasets. We can attribute this singular performance improvement of `Short noise` augmentation on the TIMIT dataset to the similar noise types (environmental noises 1) used to augment mixtures in short noise augmentation. To further see if we could leverage `Short noise` augmentation we conducted experiments by combining other augmentations with short noise augmentation. But we only observe performance improvements when `Short noise` is combined with time and frequency masking augmentation on the TIMIT dataset. Specifically, `Short noise` augmentation when combined with Frequency masking gives the most SI-SNRi improvement (1.26 dB) on the TIMIT dataset. However, it must be noted that none of the augmentations which used `Short noise` augmentation for one of the augmentation combinations leads to substantial performance improvement on LibriMix/VCTK datasets. Along similar lines combining `SpecAugment` augmentations with one or more other augmentations on average leads to deteriorated separation performance on the three datasets. Results on the combination of `SpecAugment` with `CmixDmix` are in line with results obtained with `SpecAugmented` models where `Frequency masking` tends to perform the worst.

In summary, when using a single augmentation for training, there does not seem to be any relationship between the use of source and non-source preserving augmentation and the performance of the speech separation model. Also, intra-corpus experiments indicated `Dynamic Mixing` and `CutMix` augmentation to have the joint best separation performance. Both `Dynamic mixing` and `CutMix` augmentation introduces the model to novel mixtures in each epoch as compared to other augmentations such as `SpecAugment` that apply signal transformations to the original mixture leading to a slightly different variation of the original mixture. `Dynamic mixing` and `CutMix` augmentation when combined with `Data-only Mixup` gave the best generalization performance across the three tested datasets which can be attributed to having distinctively different mixtures in the training while the augmented mixture being semantically meaningful as compared other augmentations such as `Complete Mixup`.

5.4. Training with a different model

To verify the transferability of results obtained with the DPRNN model with a separation model that uses a different model architecture we apply the augmentations to ConvTasNet (Luo and Nima Mesgarani, 2019). ConvTasNet (Luo and Nima Mesgarani, 2019) is a time domain speech separation model which uses convolutional architecture as opposed to DPRNN (Luo et al., 2020) which is based on dual-path recurrent architecture. The results from the above comparison have been depicted in Table 6 and Fig. 7. We see consistent performance improvement across the best-performing augmentations (`CmixDmix`, `CmixDoDmix`) in both models across multiple datasets.

However, we also observe inconsistent performance among multiple individual augmentations (`Time/Frequency masking`, `Gain`) which is in line with the lack of transferability of augmentations reported by Longpre et al. (2020) when they compared LSTM based networks to Transformer based networks for classification and natural language processing tasks. However as an anomaly, `Short noise` augmentation despite being largely domain-specific augmentation improves separation performance (SI-SNRi) on TIMIT dataset with ConvTasNet (0.39 dB) and DPRNN (1.31 dB) model. This can be attributed to the presence of similar noise types in the short noise subset to that of the one used by the TIMIT dataset.

Another observed pattern was a combination of task agnostic augmentations e.g (`DataOnlyMixup`, `CmixDmix`, `CmixDoDmix`) seem to lead to the best out-of-domain generalization. Here, task-agnostic augmentations refer to augmentation operations that can be applied to a training sample regardless of their domain (e.g. image/audio/language). Similar results were obtained by Longpre et al. (2020) where they reported task agnostic augmentations to have a maximal impact when the unseen test conditions might not be well represented in the preliminary training subset. Therefore, we observe the maximum SI-SNRi improvement of 1.64 dB (DPRNN), 1.40 dB (ConvTasNet) over the Un-Augmented model using `CmixDoDmix` when testing on samples from the TIMIT dataset.

Table 6
Separation metric SI-SNRi [dB] and ΔSI-SNRi [dB] measured as the performance improvement over the Un-Augmented model when using multiple augmentations to train 2 models (DPRNN (Luo et al., 2020), ConvTasNet (Luo and Nima Mesgarani, 2019)) tested on 3 datasets (LibrMix, VCTK, TIMIT).

Augmentation	Intra-corpus				Inter-corpus							
	LibriMix				VCTK				TIMIT			
	DPRNN	ConvTasNet	DPRNN	ConvTasNet	DPRNN	ConvTasNet	DPRNN	ConvTasNet	DPRNN	ConvTasNet	DPRNN	ConvTasNet
	SI-SNRi		Δ SI-SNRi		SI-SNRi		Δ SI-SNRi		SI-SNRi		Δ SI-SNRi	
Un-Augmented	12.00	11.32	0.00	0.00	10.95	9.95	0.00	0.00	7.64	6.51	0.00	0.00
Gaussian noise	12.00	11.33	0.00	0.02	10.91	9.79	−0.04	−0.16	7.52	6.83	−0.12	0.32
Gain	11.88	11.29	−0.12	−0.02	10.82	9.91	−0.13	−0.04	7.14	7.09	−0.50	0.58
Time masking	11.92	11.04	−0.08	−0.28	10.60	9.55	−0.35	−0.39	7.47	6.83	−0.17	0.32
Frequency masking	11.77	11.36	−0.23	0.04	10.68	10.22	−0.26	0.27	7.77	5.89	0.13	−0.62
Dynamic mixing	12.11	11.25	0.11	−0.07	11.08	10.19	0.13	0.25	7.55	7.07	−0.09	0.56
Short noise	12.06	11.47	0.06	0.16	11.23	10.05	0.28	0.10	8.03	7.82	0.39	1.31
Pitch shift	4.26	2.95	−7.74	−8.36	2.64	2.57	−8.30	−7.38	−1.60	−0.26	−9.24	−6.77
Time stretch	9.80	9.85	−2.15	−1.51	8.64	8.34	−2.31	−1.61	5.44	5.86	−2.20	−0.64
Complete Mixup	11.97	11.06	−0.03	−0.25	11.02	9.45	0.07	−0.50	7.96	6.24	0.32	−0.27
Data-only Mixup	12.00	11.23	0.00	−0.08	11.31	9.96	0.36	0.01	9.06	8.12	1.41	1.61
LutMix	12.11	11.38	0.11	0.06	10.84	10.30	−0.10	0.35	7.76	7.66	0.12	1.15
CmixDmix	12.54	**11.90**	0.54	**0.59**	11.63	**11.07**	0.69	**1.12**	8.27	7.93	0.63	1.42
CmixDoDmix	**12.55**	11.67	**0.55**	0.35	**11.68**	10.79	**0.73**	0.84	**9.04**	**8.15**	**1.40**	**1.64**

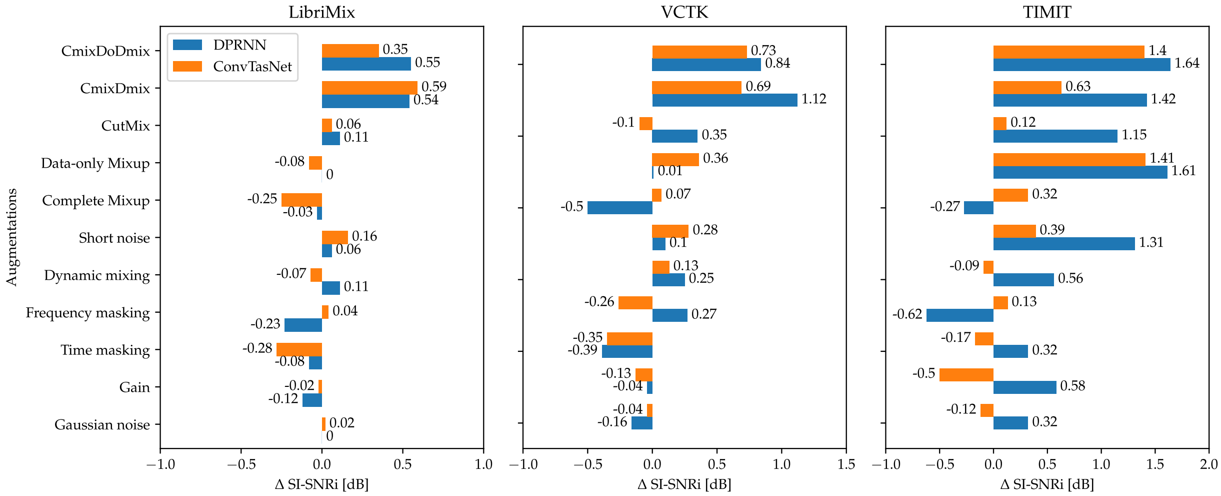


Fig. 7. Performance improvement from the Un-Augmented model (ΔSI-SNRi) using the 2 models (ConvTasNet & DPRNN) from using various augmentations for training; tested on 3 Datasets (LibriMix, VCTK, TIMIT).

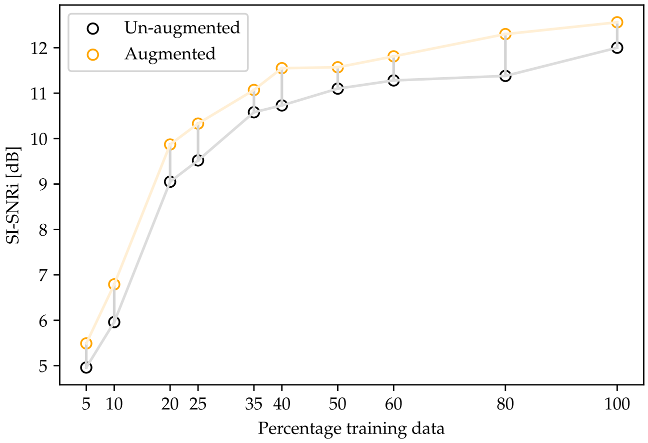


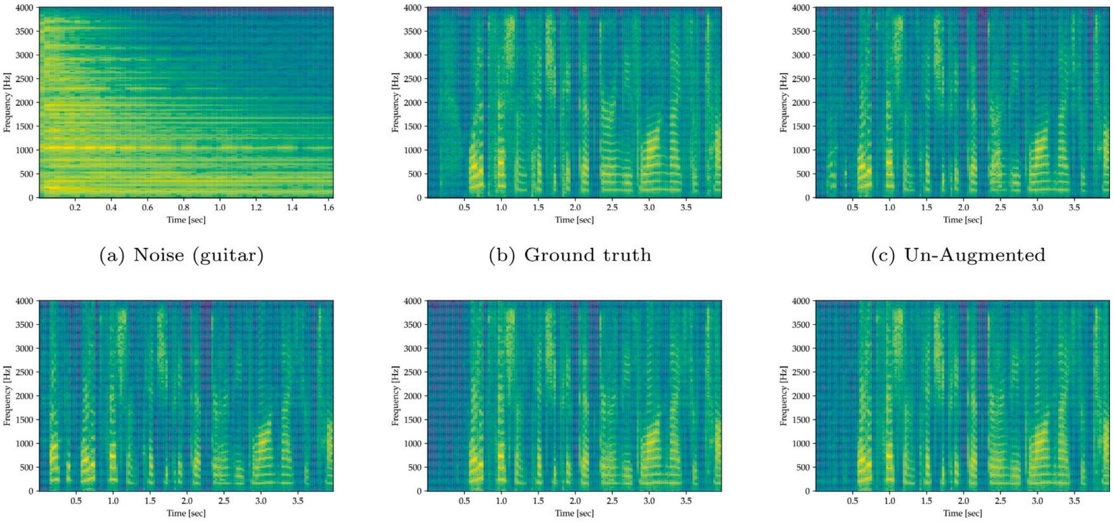
Fig. 8. SI-SNRi vs Percentage training data with CutMix + Dynamic Mixing augmentation with varying amounts of training data tested on LibriMix dataset.

5.5. Training with fewer training data

To test if the performance improvement brought about by augmentations is due to the large dataset that we have, we perform an ablation by training the DPRNN model (Luo et al., 2020) with varying amounts of training data [5, 10, 20, 25, 35, 40, 50, 60, 80, 100%] using `CmixDmix` augmentation, which is our best-performing augmentation on LibriMix dataset. The results in Fig. 8 indicate that the model trained using `CmixDmix` augmentation on average can outperform the Un-augmented model. Also, the amount of gain is very stochastic with an average improvement of 0.61 dB with a standard deviation of 0.27 dB. It can also be noted that the performance of the separation model starts to aggressively drop after the 25% data mark.

5.6. Discussion

`Data-only Mixup` performs the best among individual augmentations when tested on intra-corpus scenarios. Along similar lines, `Data-only Mixup` when combined with `CutMix` and `Dynamic mixing` (`CmixDoDmix`) lead to the best overall performance for both inter and intra-corpus testing. `Mixup` based methods have been reported to increase the robustness of the model by showing adversarial examples during training (Harris et al., 2021). Adversarial examples in our case are mixtures that are further distorted by other mixtures in the same mini-batch via linear interpolation. Additionally, `Mixup` based methods have also been reported to improve generalization by limiting the memorization ability of the model (Liang et al., 2018). On the other hand, `CutMix` unlike `Data-only Mixup`, does not semantically distort audio. `CutMix` prevents the model from overfitting on the training dataset by increasing the number of observable data points in the training sample (Harris et al., 2021). The higher number of observable



(a) Noise (guitar)

(b) Ground truth

(c) Un-Augmented

(d) CutMix

(e) Data-only Mixup

(f) CmixDoDmix

Fig. 9. Speaker separated from a mixture in the TIMIT dataset using model trained with various augmentations.

data points in our case is analogous to more speakers in a sample mixture in a mini-batch. The above assessment is supported by the quantitative results obtained using `CutMix` and `Data-only Mixup` augmentations where `CutMix` does not seem to improve separation performance on intra-corpus testing with TIMIT dataset which contains highly extreme noise conditions which are quite dissimilar to the one in WHAM noise subset used in LibriMix and VCTK dataset. However, Data-only Mixup seems robust against these noise types due to being trained on mixtures with artifacts from other mixtures in the mini-batch.

Furthermore, we analyze spectrograms (Fig. 9) of a separated speaker waveform from a mixture in the TIMIT dataset corrupted by guitar noise. Almost all the augmentations presented in Fig. 9 can get close to the spectral representation of the ground-truth speaker waveform. But separated CutMix and Un-Augmented waveforms seem to bring in minor artifacts from the other waveform at the beginning of the separated waveform. It is also evident that none of the augmentations presented in Fig. 9 are fully able to remove the noise from the separated waveform. This could perhaps be due to the nature of the noise here which resembles speech formants. Furthermore, the experimental results presented in Section 5.4 indicate that combining mutually exclusive operations of interpolation and masking from `Data-only Mixup` and `CutMix` (`CmixDoDmix`) not only improved inter and intra-corpus separation performance but also increased the robustness of the performance across the two tested models (DPRNN (Luo et al., 2020), ConvTasNet (Luo and Nima Mesgarani, 2019)).

Future work would involve developing an augmentation search policy-based method for predicting the schedule, magnitude, and probability of augmentation combination for each epoch than having fixed augmentation hyperparameters for all the epochs of training. This research can help to narrow down the search space for such augmentation policy search-based methods for speech separation and therefore bring down the training and computational resources required to identify the best augmentation strategy for training speech separation models.

In this paper we employ a simple experimental search-based strategy to determine the best augmentation combination that leads to increased generalization for speech separation in noisy environments. In addition to this, several more advanced research works have attempted to use a combination of data augmentation operations to improve the performance of machine learning models (Cubuk et al., 2019; Lim et al., 2019; Cubuk et al., 2020; Hendrycks et al., 2020; Wang et al., 2021). For instance, `AutoAugment` (Cubuk et al., 2019) trains a child network with a sequence of augmentation operations generated from a recurrent neural network (RNN). Validation accuracy from the child network is used as a reward signal to optimize the RNN to produce a better sequence of augmentations over time. `Fast AutoAugment` (Lim et al., 2019) improves on the speed of the search algorithm used to find the most effective augmentation policies by using a density matching method and splitting the training data and training each split in a distributed manner. Top-performing augmentation policies are selected from each split and a combination of best-performing policies is used to re-train the model on the entire dataset. `Adversarial AutoAugment` (Zhang et al., 2020) extended `AutoAugment` (Cubuk et al., 2019) by training a network to generate adversarial augmentation policies to make the target model more robust. It optimizes the policies directly on the target dataset instead of using smaller subsets of datasets and models, thus reducing the computational cost of training auxiliary models. `Population-Based Augmentation` (Ho et al., 2019) predicts a schedule of augmentation policies that can be applied during training over the epochs than having a fixed augmentation policy. `AugMix` (Hendrycks et al., 2020), `RandAugment` (Cubuk et al., 2020) further reduce the search space by stochastically applying a sequence of augmentations. The abovementioned methods typically require large computations to automatically search for the most efficient way to combine multiple augmentations. The design of automatic augmentation to speech separation would be an interesting research direction in the future.

6. Conclusion

We conducted a comparative study of various augmentation techniques to improve the cross-corpus generalization of two speech separation models (DPRNN (Luo et al., 2020), ConvTasNet (Luo and Nima Mesgarani, 2019)). The study evaluated ten individual augmentation methods and various combination strategies with intra-corpus and inter-corpus testing based on three speech datasets (LibriMix, TIMIT, VCTK). It was shown that while augmentation cannot significantly improve the separation performance for intra-corpus testing, it can improve the separation performance effectively for inter-corpus testing, which is the main objective of the model generalization. Among individual augmentations, `Data-only Mixup` achieves the best inter-corpus generalization performance. Among the augmentation combinations, `CmixDoDmix`, which is a combination of three augmentations, `CutMix`, `Data-only Mixup` and `Dynamic mixing`, achieves the best inter-corpus generalization performance. Both `Dynamic mixing` and CutMix augmentation exposes the model to completely new mixtures in each iteration of training than having a transformed version of the same mixture as in most augmentation operations. This stochastic introduction of new training samples seems to be the key to having better generalization across multiple datasets. It was also shown that the combined augmentation (e.g. CmixDmix as a combination of Dynamic mixing and CutMix) can improve the separation performance when the speech separation model is trained with fewer data.

In future work we will investigate the performance of the augmentation methods on more state-of-the-art speech separation models, e.g. the time-domain DPTNet model (Chen et al., 2020) and the time–frequency domain TF-Gridnet model (Wang et al., 2022). We will investigate the performance of the augmentation methods on models separating more than two overlapped speakers and working in a wider frequency range (e.g. sampling rate 16 kHz). In this paper, we empirically proposed a simple strategy to combine different augmentation methods. Future work would develop an automatic and smarter strategy for the combination of augmentation methods to further improve the generalization of the model.

CRediT authorship contribution statement

Ashish Alex: Conception and design of study, Acquisition of data, Analysis and/or interpretation of data, Writing – original draft, Writing – review & editing. **Lin Wang:** Conception and design of study, Acquisition of data, Analysis and/or interpretation of data, Writing – original draft, Writing – review & editing. **Paolo Gastaldo:** Conception and design of study, Acquisition of data, Analysis and/or interpretation of data, Writing – original draft, Writing – review & editing. **Andrea Cavallaro:** Conception and design of study, Acquisition of data, Analysis and/or interpretation of data, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Audio samples of separated mixtures with model trained using various augmentations can be found at http://www.eecs.qmul.ac.uk/~linwang/demo/augmentation.html.

Acknowledgment

All authors approved the version of the manuscript to be published.

References

Alex, A., Wang, L., Gastaldo, P., Cavallaro, A., 2021. Mixup augmentation of generalizable speech separation. In: Proc. Int. Workshop Multimedia Signal Process. pp. 1–6.

Arakawa, R., Takamichi, S., Saruwatari, H., 2019. Implementation of DNN-based real-time voice conversion and its improvements by audio data augmentation and mask-shaped device. In: Proc. ISCA Workshop Speech Synthesis. pp. 93–98.

Brown, G.J., Wang, D., 2005. Separation of speech by computational auditory scene analysis. In: Speech Enhancement. Springer, pp. 371–402.

Chen, J., Mao, Q., Liu, D., 2020. Dual-path transformer network: Direct context-aware modeling for end-to-end monaural speech separation. In: Proc. Interspeech. pp. 2642–2646.

Cosentino, J., Pariente, M., Cornell, S., Deleforge, A., Vincent, E., 2020. LibriMix: An open-source dataset for generalizable speech separation. arXiv preprint arXiv:2005.11262.

Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V., 2019. AutoAugment: Learning augmentation strategies from data. In: Proc. IEEE Conf. Computer Vision Pattern Recognition. pp. 113–123.

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

DeVries, T., Taylor, G.W., 2017. Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552.

Gao, T., Du, J., Dai, L.-R., Lee, C.-H., 2017. A unified DNN approach to speaker-dependent simultaneous speech enhancement and speech separation in low SNR environments. Speech Commun. 95, 28–39.

Garofolo, J.S., 1993. TIMIT Acoustic Phonetic Continuous Speech Corpus. Linguistic Data Consortium.

Harris, E., Marcu, A., Painter, M., Niranjan, M., Prugel-Bennett, A., Hare, J., 2021. FMix: Enhancing mixed sample data augmentation. arXiv preprint arXiv:2002.12047.

Hendrycks, D., Mu, N., Cubuk, E.D., Zoph, B., Gilmer, J., Lakshminarayanan, B., 2020. AugMix: A simple data processing method to improve robustness and uncertainty. In: Proc. Int. Conf. Learning Representations.

Hershey, J.R., Chen, Z., Le Roux, J., Watanabe, S., 2016. Deep clustering: Discriminative embeddings for segmentation and separation. In: Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. pp. 31–35.

Ho, D., Liang, E., Chen, X., Stoica, I., Abbeel, P., 2019. Population based augmentation: Efficient learning of augmentation policy schedules. In: Proc. Int. Conf. Machine Learning. pp. 2731–2741.

Inoue, H., 2018. Data augmentation by pairing samples for images classification. arXiv preprint arXiv:1801.02929.

Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Proc. Int. Conf. Machine Learning. pp. 448–456.

Kadioglu, B., Horgan, M., Liu, X., Pons, J., Darcy, D., Kumar, V., 2020. An empirical study of conv-TasNet. In: Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. pp. 7264–7268.

Kim, G., Han, D.K., Ko, H., 2021. SpecMix: A mixed sample data augmentation method for training with time-frequency domain features. In: Proc. Interspeech. pp. 546–550.

Ko, T., Peddinti, V., Povey, D., Khudanpur, S., 2015. Audio augmentation for speech recognition. In: Proc. Interspeech. pp. 3586–3589.

Kolbæk, M., Yu, D., Tan, Z.-H., Jensen, J., 2017. Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks. IEEE/ACM Trans. Audio Speech Lang. Process. 25 (10), 1901–1913.

Laroche, J., Dolson, M., 1999. Improved phase vocoder time-scale modification of audio. IEEE/ACM Trans. Audio Speech Lang. Process. 7 (3), 323–332.

Le Roux, J., Wisdom, S., Erdogan, H., Hershey, J.R., 2019. SDR–half-baked or well done? In: Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. pp. 626–630.

Lemley, J., Bazrafkan, S., Corcoran, P., 2017. Smart augmentation learning an optimal data augmentation strategy. IEEE Access 5858–5869.

Liang, D., Yang, F., Zhang, T., Yang, P., 2018. Understanding mixup training methods. IEEE Access 6, 58774–58783.

Lim, S., Kim, I., Kim, T., Kim, C., Kim, S., 2019. Fast AutoAugment. In: Proc. Advances in Neural Information Process. Systems. pp. 6665–6675.

Liu, Y., Delfarah, M., Wang, D., 2020. Deep CASA for talker-independent monaural speech separation. In: Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. pp. 6354–6358.

Liu, Y., Wang, D., 2019. Divide and conquer: A deep CASA approach to talker-independent monaural speaker separation. IEEE/ACM Trans. Audio Speech Lang. Process. 27 (12), 2092–2102.

Longpre, S., Wang, Y., DuBois, C., 2020. How effective is task-agnostic data augmentation for pretrained transformers? In: Empirical Methods in Natural Lang. Process. pp. 4401–4411.

Luo, Y., Chen, Z., Yoshioka, T., 2020. Dual-path RNN: Efficient long sequence modeling for time-domain single-channel speech separation. In: Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. pp. 46–50.

Luo, Y., Mesgarani, N., 2018. TasNet: Time-domain audio separation network for real-time, single-channel speech separation. In: Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. pp. 696–700.

Luo, Y., Mesgarani, N., 2020. Separating varying numbers of sources with auxiliary autoencoding loss. arXiv preprint arXiv:2003.12326.

Luo, Y., Nima Mesgarani, 2019. Conv-TasNet: Surpassing ideal time–frequency magnitude masking for speech separation. IEEE/ACM Trans. Audio Speech Lang. Process. 27 (8), 1256–1266.

Maciejewski, M., Wichern, G., McQuinn, E., Le Roux, J., 2020. WHAMR!: Noisy and reverberant single-channel speech separation. In: Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. pp. 696–700.

Michelsanti, D., Tan, Z.-H., Zhang, S.-X., Xu, Y., Yu, M., Yu, D., Jensen, J., 2021. An overview of deep-learning-based audio-visual speech enhancement and separation. IEEE/ACM Trans. Audio Speech Lang. Process. 29, 1368–1396.

Mukhutdinov, D., Alex, A., Cavallaro, A., Wang, L., 2023. Deep learning models for single-channel speech enhancement on drones. IEEE Access 11, 22993–23007.

Nachmani, E., Adi, Y., Wolf, L., 2020. Voice separation with an unknown number of multiple speakers. In: Proc. Int. Conf. Machine Learning. pp. 7164–7175.

Pariente, M., Cornell, S., Cosentino, J., Sivasankaran, S., Tzinis, E., Heitkaemper, J., Olvera, M., Stöter, F.-R., Hu, M., Martín-Doñas, J.M., Ditter, D., Frank, A., Deleforge, A., Vincent, E., 2020a. Asteroid: The PyTorch-based audio source separation toolkit for researchers. In: Proc. Interspeech. pp. 2637–2641.

Pariente, M., Cornell, S., Deleforge, A., Vincent, E., 2020b. Filterbank design for end-to-end speech separation. In: Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. pp. 6364–6368.

Park, D.S., Chan, W., Zhang, Y., Chiu, C.-C., Zoph, B., Cubuk, E.D., Le, Q.V., 2019. SpecAugment: A simple data augmentation method for automatic speech recognition. In: Proc. Interspeech. pp. 2613–2617.

Salamon, J., Bello, J.P., March 2017. Deep convolutional neural networks and data augmentation for environmental sound classification. IEEE Signal Process. Lett. 24 (3), 279–283.

Schmidt, M.N., Olsson, R.K., 2006. Single-channel speech separation using sparse non-negative matrix factorization. In: Proc. Interspeech. pp. 2614–2617.

Shorten, C., Khoshgoftaar, T.M., 2019. A survey on image data augmentation for deep learning. J. Big Data 6 (1), 1–48.

Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., Webb, R., 2016. Learning from simulated and unsupervised images through adversarial training. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition. pp. 2107–2116.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: A simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15 (1), 1929–1958.

Subakan, C., Ravanelli, M., Cornell, S., Bronzi, M., Zhong, J., 2021. Attention is all you need in speech separation. In: Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. pp. 21–25.

Summers, C., Dinneen, M.J., 2019. Improved mixed-example data augmentation. In: Proc. IEEE Winter Conf. Applications Computer Vision. pp. 1262–1270.

Tokozume, Y., Ushiku, Y., Harada, T., 2018a. Between-class learning for image classification. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition. pp. 5486–5494.

Tokozume, Y., Ushiku, Y., Harada, T., 2018b. Learning from between-class examples for deep sound recognition. In: Proc. Int. Conf. Learning Representations.

Veaux, C., Yamagishi, J., MacDonald, K., 2016. Superseded-CSTR VCTK Corpus: English Multi-Speaker Corpus for CSTR Voice Cloning Toolkit. The Centre for Speech Technology Research, University of Edinburgh.

Wang, L., 2014. Multi-band multi-centroid clustering based permutation alignment for frequency-domain blind speech separation. Digit. Signal Process. 31, 79–92.

Wang, L., Cavallaro, A., 2018. Pseudo-determined blind source separation for ad-hoc microphone networks. IEEE/ACM Trans. Audio Speech Lang. Process. 26 (5), 981–994.

Wang, L., Cavallaro, A., 2020. Deep learning assisted time-frequency processing for speech enhancement on drones. IEEE Trans. Emerg. Top. Comput. Intell. 5 (6), 871–881.

Wang, D., Chen, J., 2018. Supervised speech separation based on deep learning: An overview. IEEE/ACM Trans. Audio Speech Lang. Process. 26 (10), 1702–1726.

Wang, Z.-Q., Cornell, S., Choi, S., Lee, Y., Kim, B.-Y., Watanabe, S., 2022. TF-GridNet: integrating full-and sub-band modeling for speech separation. arXiv preprint arXiv:2211.12433.

Wang, Z.-Q., Le Roux, J., Hershey, J.R., 2018a. Alternative objective functions for deep clustering. In: Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. pp. 686–690.

Wang, Z.-Q., Roux, J.L., Wang, D., Hershey, J.R., 2018b. End-to-end speech separation with unfolded iterative phase reconstruction. arXiv preprint arXiv:1804.10204.

Wang, H., Xiao, C., Kossaifi, J., Yu, Z., Anandkumar, A., Wang, Z., 2021. AugMax: Adversarial composition of random augmentations for robust training. In: Proc. Neural Information Process. Systems.

Wei, S., Xu, K., Wang, D., Liao, F., Wang, H., Kong, Q., 2018. Sample mixed-based data augmentation for domestic audio tagging. In: Proc. Workshop Detection and Classification of Acoustic Scenes and Events. pp. 93–97.

Wei, S., Zou, S., Liao, F., et al., 2020. A comparison on data augmentation methods based on deep learning for audio classification. J. Phys. Conf. Ser. 1453 (1), 012085.

Wichern, G., Antognini, J., Flynn, M., Zhu, L.R., McQuinn, E., Crow, D., Manilow, E., Le Roux, J., 2019. WHAM!: Extending speech separation to noisy environments. In: Proc. Interspeech. pp. 1368–1372.

Williamson, D.S., Wang, Y., Wang, D., 2015. Complex ratio masking for monaural speech separation. IEEE/ACM Trans. Audio Speech Lang. Process. 24 (3), 483–492.

Wu, Y.-K., Huang, K.-P., Tsao, Y., Lee, H.-y., 2021. One shot learning for speech separation. In: Proc. Int. Conf. Acoust., Speech, Signal Process. pp. 5769–5773.

Wu, Y.-K., Tuan, C.-I., Lee, H.-Y., Tsao, Y., 2020. SADDEL: Joint speech separation and denoising model based on multitask learning. arXiv preprint arXiv:2005.09966.

Xu, Y., Du, J., Dai, L.-R., Lee, C.-H., 2015. A regression approach to speech enhancement based on deep neural networks. IEEE/ACM Trans. Audio Speech Lang. Process. 23 (1), 7–19.

Yamagishi, J., Veaux, C., MacDonald, K., 2019. CSTR VCTK Corpus: English Multi-Speaker Corpus for CSTR Voice Cloning Toolkit. The Centre for Speech Technology Research, University of Edinburgh.

Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y., 2019. Cutmix: Regularization strategy to train strong classifiers with localizable features. In: Proc. IEEE/CVF Int. Conf. Computer Vision. pp. 6023–6032.

Zagoruyko, S., Komodakis, N., 2016. Wide residual networks. In: Proc. British Machine Vision Conf. pp. 87.1–87.12.

Zeghidour, N., Grangier, D., July 2021. Wavesplit: End-to-end speech separation by speaker clustering. IEEE/ACM Trans. Audio, Speech, and Lang. Process. 2840–2849.

Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O., 2017. Understanding deep learning requires rethinking generalization. In: Proc. Int. Conf. Learning Representations. pp. 107–115.

Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D., 2018a. Mixup: Beyond empirical risk minimization. In: Proc. Int. Conf. Learning Representations.

Zhang, X., Wang, Q., Zhang, J., Zhong, Z., 2020. Adversarial autoaugment. In: Proc. Int. Conf. Learning Representations.

Zhang, H., Zhang, X., Gao, G., 2018b. Training supervised speech separation system to improve STOI and pesq directly. In: Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. pp. 5374–5378.

Zhong, Z., Zheng, L., Kang, G., Li, S., Yang, Y., 2020. Random erasing data augmentation. In: Proc. Conf. Artificial Intelligence. pp. 13001–13008.