An Empirical Study of Training Mixture Generation Strategies on Speech Separation: Dynamic Mixing and Augmentation

Shukjae Choi[1,2], Younglo Lee[1,2], Jihwan Park[1,2], Hyung Yong Kim[1,2], Byeong-Yeol Kim[1,2],
Zhong-Qiu Wang[3], Shinji Watanabe[3]
[1] Hyundai Motor Company, [2] 42dot.ai, Seoul, Republic of Korea
E-mail: {shukjae.choi, younglo.lee, jihwan.park, hyungyong.kim, byeongyeol.kim}@42dot.ai
[3] Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, USA
E-mail: wang.zhongqiu41@gmail.com, shinjiw@cmu.edu

Abstract—**Deep learning has dramatically advanced speech separation (SS) in the past decade. Although advances in model architectures play an essential role in improving the separation performance, an efficient training strategy is also important. In this study, we investigate various strategies for training mixture generation in SS, considering that such strategies are likely essential in improving the generalization abilities of the trained models. More specifically, instead of using the vanilla training mixtures pre-generated by a given dataset, we remix clean source signals to generate more mixtures by using dynamic mixing (DM), which is an on-the-fly speech mixing strategy for model training. In addition, we combine DM with other data augmentation methods to further improve the separation performance. We analyze the effects of training data generation strategies for training sets at different scales and with various diversities. Evaluation results on multiple public datasets suggest that increasing the number of speech mixtures using DM with data augmentations is a very effective strategy for SS, especially for training sets with a limited number of clean sources.**

I. Introduction

Riding on the tide of deep learning, significant progress has been made in speech separation (SS) [1]. Since deep clustering [2], [3] and permutation invariant training (PIT) [4], [5] successfully solved the label permutation problem in talker-independent speaker separation, many subsequent studies have been focusing on designing more efficient and more end-to-end neural networks to improve the performance. Representative models along this line include time-domain audio separation network (TasNet) [6], fully-convolutional TasNet (Conv-TasNet) [7], dual-path recurrent neural network (DPRNN) [8], dual-path transformer network [9], Wavesplit [10] and SepFormer [11], which have shown remarkable separation performance and strong potential towards solving the SS problem.

Although architectural advances play an essential role in improving separation, they are often simultaneously introduced with changes in the details of the training methodologies, hyper-parameters, or data augmentation techniques [12]. This paper studies training mixture generation strategies for SS. We investigate how dynamic mixing (DM) and various data augmentation methods influence the performance of SS. This investigation is necessary because, even for advanced SS architectures, an effective strategy for training data generation is needed for the commercialization of SS systems.

Many on-the-fly data augmentation techniques have been proposed in various audio signal processing tasks [13], [14]. In automatic speech recognition (ASR) or speech enhancement, various data augmentations can be easily applied on the fly to a clean speech source [15], [16], [17]. On-the-fly data augmentation can prevent the model from overfitting and make it more robust to various situations by imitating various environmental conditions. It can usually improve the robustness of modern deep neural network (DNN) models.

In this paper, we use the term “DM” to refer to a method for creating speech mixtures and the term “data augmentation” for speech source perturbations. Since SS tasks aim to separate speech signals from the mixture, we can easily presume that DM can control the complexity of the training mixtures and would directly affect model performance. However, in SS, speech mixtures are conventionally fixed and used in the training phase for the standard SS benchmark, so an application of DM and data augmentation is rarely investigated. For a general performance improvement in SS, it would be beneficial to investigate the effect of DM and various augmentations. In the SS literature, Wavesplit [10] first revealed the effectiveness of DM. Subsequently, SepFormer [11] achieved the state-of-the-art separation performance on the WSJ0-2mix dataset [2] by combining a DM strategy with other types of data augmentation. Different from the above studies, our study concentrates on the effects of DM and augmentation itself. We analyzed the benefits of using DM on datasets with various scales and speaker diversities and compared them with their vanilla cases, where no DM is applied. We also thoroughly investigated the effectiveness of each individual augmentation type as well as their combinations for SS. During the investigation, we propose a phase-shifting data augmentation method that has the potential to be applied to a variety of speech processing tasks.

We chose Conv-TasNet [7] as the major reference model to analyze the effects of DM and augmentations. DPRNN [8] is additionally selected to show that a similar trend in perfor-

TABLE I
NUMBER OF MIXTURES, SPEAKERS, AND SOURCE SIGNALS FOR TRAINING IN LIBRI2MIX TRAIN-$\{360, 100\}$ AND WHAM!.

Datasets	#mixtures	#speakers	#sources
Libri2Mix train-360	50,800	921	101,600
Libri2Mix train-100	13,900	251	27,800
WHAM! train	20,000	101	8,769

TABLE II
NUMBERS OF MIXTURES, SPEAKERS, AND SOURCE SIGNALS IN VCTK2MIX, LIBRI2MIX AND WHAM! TEST SETS.

Datasets	#mixtures	#speakers	#sources
VCTK2Mix$_{\text{trim}}$	3,000	108	6,000
Libri2Mix	3,000	40	2,073
WHAM!	3,000	18	1,770

mance can also be observed in other models. Both models are commonly-used baselines in many recent studies [9], [10], [11], [18]. For the training data, WHAM! [19] and Libri2Mix [20] were used considering their popularity. Their test sets and the VCTK2Mix test set [20], which cover a wide range of scales in terms of the numbers of speakers and source signals, were chosen to analyze the performance in matched and mismatched conditions. We picked these datasets because they contain realistic noisy mixtures for model training, which are important for product development.

The rest of this paper is organized as follows. We first describe our experimental design in Section II and then detail the proposed training strategies in Section III, followed by experiments and discussions in Section IV. Section V concludes this paper.

II. BACKGROUND

This section describes the models, metrics, and datasets we use to show the effectiveness of our training strategies.

A. Reference models: Conv-TasNet and DPRNN

Since TasNet [6] was proposed to mitigate the inherent phase estimation problem of the short-time Fourier transform (STFT) domain, real-valued masking-based separation methods, Conv-TasNet [7] has become a milestone in the history of time-domain SS. It outperformed previous STFT-domain separation models and even surpassed several oracle magnitude-domain masks. This model effectively increases the size of the receptive field by stacking dilated convolutions to model long-range contextual information while keeping the number of parameters moderate by using depth-wise separable convolutions.

Although Conv-TasNet achieved superior results over conventional approaches, convolutions with fixed receptive fields may have difficulty in learning extremely long-range temporal dependencies and hence may limit the separation performance. DPRNN [8] was proposed to address the above problem by using recurrent neural network (RNN) layers at multiple time scales. It splits a long input sequence into shorter chunks and processes them locally and globally with intra- and inter-block RNNs. This makes the modeling of extremely long-range temporal contexts possible [21].

These two models have been widely used as baselines in SS [9], [10], [11], [18]. We selected these models to show that our training strategies are not limited to a particular model and are general techniques that can bring improvements to many different models.

B. Metrics

Scale-invariant signal-to-distortion ratio (SI-SDR) [22] is a commonly used evaluation metric in SS. All of our models are trained with PIT [5] to maximize the SI-SDR between target and estimated sources. For the comparison metric, we used SI-SDR improvement (ΔSI-SDR), which is the difference between the SI-SDRs of the estimated sources and the mixture.

C. Dataset

There are many open datasets for SS. The most popular one is WSJ0-2mix [2], which has been commonly used to validate the effectiveness of DNN architectures. If the goal is to perform separation in realistic noisy environments, WHAM! [19], WHAMR! [23], and Libri2Mix [20] are commonly used. Among the public SS datasets, we selected the WHAM! and the Libri2Mix, both containing noisy two-speaker mixtures, to validate our proposed strategies. Both datasets use the WHAM! noises [19] as the noise sources, but the signal-to-noise ratio (SNR) of WHAM! is slightly lower on average. In addition, the two datasets differ in the number of speakers and clean source signals in their training sets, and the WHAM! training mixtures are generated based on a relatively smaller set of speakers and clean source signals (see Table I for the details).

Evaluation results serve as an indicator of whether the trained models would perform well after being deployed in real-world conditions. Besides the WHAM! and Libri2Mix test sets, we additionally evaluate our models using the VCTK2Mix test set, which contains a larger set of speakers (see Table II), to show their cross-corpus generalization abilities. We applied two pre-processings to the sources of the VCTK2Mix test set. First, we further trimmed the silence periods in the VCTK2Mix sources because ΔSI-SDR is reported to be less meaningful in silence periods [24], and long silence periods in the VCTK2Mix sources are already known and dealt with in [20]. We also found that some VCTK2Mix source signals start without pauses, unlike the other source signals in the training and test sets we selected. For the consistency of the evaluation results, zero padding with a length sampled from $\mathcal{U}(0.5, 0.7)$ seconds was applied at the beginning of the VCTK2Mix source signals, where $\mathcal{U}(a, b)$ denotes the uniform distribution between a and b. Then, we generated VCTK2Mix$_{\text{trim}}$ test sets, denoted as VCTK2Mix for simplicity in the rest of the article.

III. TRAINING STRATEGY

This section describes the strategies for training mixture generation, including DM and various types of data augmentations. The conditions and settings of the training strategies for experiments are also introduced.

A. Dynamic mixing

DM can be regarded as a type of data augmentation, often referred to as on-the-fly data augmentation in other speech processing tasks. For example, clean speech is often perturbed with noise signals on the fly when training ASR [15] or enhancement models [16], [17]. However, in the context of SS, we restrict the term "dynamic mixing" to on-the-fly multi-speaker mixture generation. That is, the clean speech sources in the fixed, pre-generated training set are randomly chosen and mixed on the fly to generate mixtures. So far, only a few studies employ the DM strategy for SS [10], [11].

Several factors need to be considered when utilizing DM. Since new speaker mixtures are continuously generated, the models can observe more speaker and source combinations than by using the vanilla pre-defined training set. As a result, the models need more training iterations than when using the vanilla training set, if the same training setup is used.

To fairly compare the model trained with dynamically mixed signals and the one trained with the vanilla pre-defined mixtures, we feed the models the same number of mixtures in each epoch, which equals the number of mixtures in the pre-defined training set.

In practice, we can use DM to generate mixtures with very different characteristics from the vanilla, pre-defined mixtures. Because our focus is on comparing the DM with the vanilla in terms of mixture diversity, we decide to closely follow the mixing policy of the vanilla dataset. When performing DM on the WHAM! training set, we sampled the relative energy levels among the two speakers and the noise from the energy-level distribution of the pre-defined WHAM! mixtures for each newly generated mixture. This method is similar to the DM method implemented in the SpeechBrain toolkit [25]. We directly applied the mixing policy in [20] to the LibriSpeech clean sources when using DM on Libri2Mix train-{360, 100}. It assigns an absolute range of random levels in the loudness unit relative to full scale (LUFS) to speech and noise signals.

B. Data augmentation

Data augmentation applies various modifications to source signals, imitating various environmental conditions to increase the complexity and diversity of training data. It can prevent the model from overfitting and make it more robust to various situations. Depending on the task type, different augmentation methods should be applied. In classification tasks like ASR or keyword spotting, latent representation of data sources is used, making many types of augmentations like removal [26], mixing [27], or transposition [28] possible. Numerous studies have demonstrated the benefits of these types of augmentations. However, not all types of augmentations can be applied to a regression task because it needs to accurately estimate each sample of the source signal. As a result, for regression tasks, data augmentation that preserves the information from the source is more effective, such as speed, tempo, and pitch shift.

In SS, many studies usually only use the pre-defined training set for model training so that they can fairly compare their models with earlier models that were trained on the same set of mixtures. When building practical products, it is often desirable to leverage data augmentations for better separation. Recently, SepFormer [11] observed clear improvements by combining speed augmentation and DM. Motivated by their findings, we comprehensively investigated the effects of using other types of data augmentations with DM.

Given a clean-speech segment, the following augmentations were considered:

- "**pitch**," "**tempo**," and "**speed**" augmentations are probably the most popular augmentations used in speech signal processing since they preserve speech characteristics and can, to some extent, increase diversity. "**pitch**" shifts the pitch of the segment without changing the tempo, whereas "**tempo**" stretches the segment in time without changing the pitch. "**speed**" adjusts the segment speed, affecting both pitch and tempo by first changing the sampling rate information and then re-sampling the segment back to the original sample rate. In "**pitch**," the pitch is shifted by $\mathcal{U}(-3, 3)$ semitones, which is the distance between two adjacent notes in music. The scales of "**tempo**" and "**speed**" are chosen from the set $\{0.9, 1.0, 1.1\}$ [29].
- "**gain**" and "**white noise (wn)**" simulate actual recording conditions while preserving the original sources. "**gain**" alters the overall level of the mixture and target signals by a factor uniformly sampled from the range $[-10, 10]$ dB. This augmentation can make the trained models more robust to variations in input gains. "**wn**" applies a white noise with a LUFS sampled from $\mathcal{U}(-90, -46)$ dB. This augmentation could alleviate overfitting in noisy environments.
- In monaural audio signals, human ears are known to be insensitive to the phase of the sound [30]. If we change the phase of the sound, the segment in the temporal and phase domain will be changed, but the spectral magnitude of the segment and the audible sound will remain the same. Because of these characteristics, we believe that "**phase shift**" augmentation has the potential to be applied to a variety of machine learning models operating in temporal, phase, or complex domains. "**phase shift**" shifts the phase of the segment by a value θ sampled uniformly from the range $[-\pi, \pi)$ (i.e., the segment is multiplied by $e^{j\theta}$, where j denoting the imaginary unit, in the STFT domain). It is an extension of "**polarity inversion**," which multiplies -1 to the segment and shifts the phase by π.
- In classification tasks such as ASR, a model is often trained to be able to guess the original sources in latent space, which makes it reasonable to use augmentations that corrupt the original sources, such as removal or transposition. "**drop chunk**" and "**drop frequency**" are components proposed in SpecAugment [26]. They are

TABLE III
AVERAGE ΔSI-SDR (DB) RESULTS OF DM AND FIXED-MIXTURE CASES (VANILLA).

Training set		Conv-TasNet						DPRNN					
		WHAM!		**Libri2Mix**		**VCTK2Mix**		**WHAM!**		**Libri2Mix**		**VCTK2Mix**	
		DM	vanilla	DM	vanilla	DM	vanilla	DM	vanilla	DM	vanilla	DM	vanilla
Libri2Mix	100%	**13.14**	12.77	**12.59**	12.23	**11.76**	11.10	**14.10**	13.65	**13.27**	12.89	**12.57**	11.67
train-360	25%	**12.06**	11.43	**11.52**	10.87	**10.43**	9.68	**12.87**	12.72	**12.07**	11.95	**11.27**	10.60
Libri2Mix	100%	**11.90**	11.51	**11.45**	11.12	**10.44**	9.54	**12.82**	12.64	**12.07**	11.97	**11.20**	10.62
train-100	25%	**10.43**	9.51	**10.06**	9.27	**8.77**	7.42	**11.40**	10.52	**10.85**	10.23	**9.67**	8.29
WHAM!	100%	**13.88**	13.09	**8.72**	8.47	**8.64**	7.92	**14.49**	13.84	**9.33**	8.72	**9.49**	8.73
train	25%	**12.47**	4.72	**7.52**	4.65	**7.67**	-2.79	**13.16**	4.69	**8.20**	5.34	**8.31**	-4.70

well-known for their effectiveness and simplicity. "**drop chunk**" randomly removes signals in the time domain, and "**drop frequency**" randomly removes some frequency bands in the frequency domain. We included the "**reverse**" augmentation [28], following its success in ASR. We randomly split each utterance into segments with lengths drawn from $\mathcal{U}(5, 10)$ ms and flipped the samples in the time domain.

We also investigated various combinations of the above augmentations to achieve better separation. After performing augmentations to the clean-speech segment of each speaker, we mix them according to the mixing policy to generate a mixture for model training.

IV. Experiments and Discussion

For fair comparisons, all experiments were performed with the same hyper-parameters for each model. For Conv-TasNet [7], the training parameters were a batch size of 16 and an initial learning rate of 0.001, which were multiplied by 0.98 every two epochs. The mixture length was 2.8 seconds during training. For DPRNN, we used a kernel size of 16, a batch size of 16, and a chunk size of 150 frames, and the mixture length during training was 4.8 seconds. All models were trained in the same sampling rate of 16 kHz. We matched the number of mixtures generated in DM cases with the number of mixtures of the corresponding vanilla set in each epoch. To match the number of total mixtures fed to the models, the models trained for the DM and vanilla cases were evaluated when the model trained for the vanilla case was converged.

A. DM vs. vanilla

This section compares the SS performance in the DM and the fixed-mixture (i.e., vanilla) cases. For both cases, we trained Conv-TasNet and DPRNN using the Libri2Mix train-$\{360, 100\}$ or the WHAM! training set, and reported their ΔSI-SDR results on the WHAM!, Libri2Mix, and VCTK2Mix test sets. The evaluation results are shown in Table III. The same numbers of speakers, sources, and mixtures for each epoch were used when training using 100% of each dataset. The ΔSI-SDR in the DM cases was always higher than that in the vanilla cases for all the training set cases and for both models. This aligns with the findings of earlier studies such as Wavesplit [10] or SepFormer [11], which found that because DM generates more speaker mixtures on the fly, DNN models can learn from many more speaker combinations and yield better results. We can observe that the performance on VCTK2Mix was relatively closer to other test sets than in the previous studies [20], [31]. It is because we trimmed VCTK2Mix sources as mentioned in Section II-C and zero-padded them to have a consistent beginning, and hence the evaluation became more reliable.

TABLE IV
AVERAGE ΔSI-SDR (DB) RESULTS WHEN APPLYING VARIOUS AUGMENTATIONS. NUMBERS IN PARENTHESIS ARE DIFFERENCE FROM EACH TEST SET'S DYNAMIC MIXING CASE. "ALL AUGS." MEANS ALL AUGMENTATIONS ARE APPLIED.

Augmentations	**WHAM!**	**Libri2Mix**	**VCTK2Mix**
Dynamic mixing	13.88	8.72	8.64
+pitch	**14.11 (0.23)**	9.28 (0.56)	9.39 (0.75)
+tempo	13.82 (-0.06)	8.82 (0.10)	8.66 (0.02)
+speed	13.98 (0.10)	9.24 (0.52)	9.37 (0.73)
+phase shift (ps)	13.66 (-0.22)	9.94 (1.22)	9.84 (1.20)
+polarity inversion	14.05 (0.17)	9.86 (1.14)	9.59 (0.95)
+gain	13.90 (0.02)	8.74 (0.02)	8.47 (-0.17)
+white noise (wn)	13.81 (-0.07)	8.83 (0.11)	8.69 (0.05)
+drop chunk (dc)	13.72 (-0.16)	8.75 (0.03)	8.70 (0.06)
+drop frequency	13.74 (-0.14)	9.40 (0.68)	9.06 (0.42)
+reverse	13.30 (-0.58)	8.51 (-0.21)	8.47 (-0.17)
+pitch+tempo	14.04 (0.16)	9.37 (0.65)	9.59 (0.95)
+pitch+ps	13.52 (-0.36)	9.97 (1.25)	9.92 (1.28)
+pitch+tempo+ps	13.63 (-0.25)	**10.10 (1.38)**	**10.05 (1.41)**
+gain+dc+wn	13.77 (-0.11)	8.74 (0.02)	8.80 (0.16)
+all augs.	12.49 (-1.39)	9.53 (0.81)	8.93 (0.29)

We also simulated cases when the original training set is small because the number of training examples is usually limited in many real-world applications. In such a scenario, DM becomes very useful because it can continuously produce new mixtures for training. We simulated a smaller-scale training set by reducing the number of mixtures in the vanilla cases to 25%. For the DM cases, we generated new mixtures from source signals and set the number of mixtures to generate in each epoch equal to the number of training mixtures in the vanilla case. Note that although we fed the same number of mixtures for each epoch for both DM and vanilla, the number of sources for the WHAM! 25% case were different by 2,192 and 5,154, respectively. As mentioned earlier in Section II-C, this difference is because the WHAM! reused source signals when generating mixtures. When using less training data, we observed that ΔSI-SDR degraded more in the vanilla case than in the DM case. For example, in the WHAM! train 25% case, both Conv-TasNet and DPRNN trained in the vanilla case nearly failed to learn to separate (see the "vanilla" columns in the last row of Table III), largely because there are much fewer

TABLE V
Comparison of average ΔSI-SDR (dB) results of DM with augmentations (DM+augs), DM and vanilla.

Model	**WHAM!** training set	**WHAM!** DM+augs	DM	vanilla	**Libri2Mix** DM+augs	DM	vanilla	**VCTK2Mix** DM+augs	DM	vanilla
Conv-TasNet	100%	13.63	**13.88**	13.09	**10.10**	8.72	8.47	**10.05**	8.64	7.92
	25%	10.76	**12.47**	4.72	**9.82**	7.52	4.65	**8.46**	7.67	-2.79
DPRNN	100%	14.34	**14.49**	13.84	**10.83**	9.33	8.72	**10.38**	9.49	8.73
	25%	**13.21**	13.16	4.69	**9.73**	8.20	5.34	**9.40**	8.31	-4.70

mixtures and speakers for model training. However, when trained using DM, the models trained in the 25% case showed only slightly worse performance compared to the 100% case.

These results suggest the effectiveness of DM for modern SS models, especially when the training set is small-scale, which is typically the case in many real-world applications.

B. Effectiveness of data augmentations

This section evaluates the performance of data augmentations described in Section III-B and explores their combinations to find the most effective combination for SS. We applied data augmentations and DM on the WHAM! training set to Conv-TasNet using the same parameter used in Section IV-A. When multiple augmentations are combined, each augmentation is applied with a probability of 50%. For a fair comparison, different random states were used for DM and augmentations; that is, while the stream of mixture input remained the same for all cases, only combinations of augmentations applied were changed.

The evaluation results of each augmentation for each test set are reported in Table IV. We observed that not all augmentations were effective. In addition, augmentations were effective on mismatched datasets, Libri2Mix and VCTK2Mix. Since the WHAM! test set shares relevant recording conditions, speaking styles, and mixing policy with its training set, we cannot regard the WHAM! test set as a fully mismatched set. This is likely the reason why we did not observe performance improvement.

We focused on augmentations that preserve the original information, such as stretching and shifting. The “pitch” augmentation was effective on all three test sets. We investigated the combinations of “speed,” “tempo,” and “pitch” augmentations, as they are closely related. We found that even the “speed” augmentation changes time and frequency dependently, “pitch + tempo” worked better than just using speed in all test sets. The “phase shift (ps)” and “polarity inversion” augmentations produced noticeable improvements; effectiveness was valid even when they were combined with other augmentations, indicating their broad application potential in speech signal processing. Additional experiments on combinations of augmentations suggested that “pitch + tempo + ps” was the most effective combination with a relatively small number of augmentations. This could be because the “pitch” augmentation can increase the speaker diversity, and the “tempo” and “phase shift” augmentations can increase the diversity of temporal characteristics.

We also investigated combinations of ineffective augmentations. The intention was to check if a combination of the ineffective ones would improve performance. Unlike in the ASR task, the “reverse” augmentations was ineffective for SS, likely because, in SS, detailed speech signal characteristics need to be reconstructed through regression. This observation indicates that augmentations with removal or transposition of original information are unlikely to work on regression tasks. The “gain” effect might have been canceled during the normalization process of the model. The effect of “drop chunk (dc)” overlapped with the source signal’s silence period. The “wn” augmentation was ineffective because the noise signal canceled the white noise. Since performance does not improve even if all augmentations (i.e., all augs.) are used or ineffective augmentations are combined (i.e., gain + dc + wn), individual performance needs to be carefully investigated before application.

TABLE VI
Comparison of average ΔSI-SDR (dB) results on SepFormer.

Training strategy	**WHAM!**	**Libri2Mix**	**VCTK2Mix**
vanilla	20.08	14.57	13.16
DM + speed	**21.83**	16.16	15.50
DM + augs	21.34	**17.14**	**16.15**

C. Application of augmentations to SS models

Next, we applied the best augmentation combination (i.e., pitch + tempo + phase shift) to train Conv-TasNet and DPRNN models based on the WHAM! training set and evaluated their performance on the test sets of WHAM!, Libri2Mix and VCTK2Mix. See Table V for the results.

We can see that the proposed augmentation combination is effective on mismatched datasets for both Conv-TasNet and DPRNN (see the Libri2Mix and VCTK2Mix columns). One notable finding is that even just 25% of sources with DM and augmentations showed similar or even better performance than vanilla cases and DM cases. Still, as can be seen from Section IV-B, we did not observe dramatic improvement in the matched dataset, the WHAM! test set. It is reasonable to believe that the chosen combination of augmentation increases speaker diversity and improves performance on mismatched conditions.

To verify our ideas, we applied the training strategy to a state-of-the-art SS model, SepFormer [11]. For a fair comparison, we carefully followed the configuration in the paper [11]. Two major factors of the experimental setup were different from our previous setup: the sampling rate was at 8 kHz, and the model was trained using WSJ0-2mix dataset, which is the same to WHAM! but without the noise sources. The evaluation results are shown in Table VI. The "vanilla" and "DM + speed" are the reproduced results following [11]. They show similar results to the original paper. Our proposed augmentation combination, which is "pitch + tempo + phase shift" (denoted as DM + augs in the table), was applied the same way as how the previous augmentations were applied. The results show that our ideas are consistent with the recent model, even with different SS settings.

V. Conclusion

We have analyzed the effects of training mixture generation strategies using DM and various data augmentations for SS based on popular public datasets such as WHAM! and Libri2Mix. We found that DM, which generates speech mixtures on the fly, is effective for SS. It is particularly powerful than using a conventional pre-defined mixture dataset for training, especially when the number of speakers and sources for training is at a small scale. In addition, among various augmentation methods, the ones that increase speaker diversity and temporal characteristics, such as phase, pitch, or tempo shift, are more effective. Overall, data augmentation was found effective in most cases and was more effective when the training data has limited speaker diversity and when the test set is mismatched from the training data. The effectiveness has been demonstrated using popular separation models, including Conv-TasNet and DPRNN. We believe that the findings in this paper could play an essential role in the realization of SS in consumer products.

References

[1] D. Wang and J. Chen, "Supervised speech separation based on deep learning: An overview," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 10, pp. 1702–1726, 2018.

[2] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Proc. ICASSP*, 2016, pp. 31–35.

[3] Z.-Q. Wang, J. L. Roux, and J. R. Hershey, "Alternative objective functions for deep clustering," in *Proc. ICASSP*, 2018, pp. 686–690.

[4] D. Yu, M. Kolbæk, Z.-H. Tan, and J. Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *Proc. ICASSP*, 2017, pp. 241–245.

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

[6] Y. Luo and N. Mesgarani, "TasNet: Time-domain audio separation network for real-time, single-channel speech separation," in *Proc. ICASSP*, 2018, pp. 696–700.

[7] ——, "Conv-TasNet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 8, pp. 1256–1266, 2019.

[8] Y. Luo, Z. Chen, and T. Yoshioka, "Dual-path RNN: Efficient long sequence modeling for time-domain single-channel speech separation," in *Proc. ICASSP*, 2020, pp. 46–50.

[9] J. Chen, Q. Mao, and D. Liu, "Dual-Path Transformer Network: Direct context-aware modeling for end-to-end monaural speech separation," in *Proc. Interspeech*, 2020, pp. 2642–2646.

[10] N. Zeghidour and D. Grangier, "Wavesplit: End-to-end speech separation by speaker clustering," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 2840–2849, 2021.

[11] C. Subakan, M. Ravanelli, S. Cornell, M. Bronzi, and J. Zhong, "Attention is all you need in speech separation," in *Proc. ICASSP*, 2021, pp. 21–25.

[12] I. Bello, W. Fedus, X. Du, E. D. Cubuk, A. Srinivas, T.-Y. Lin, J. Shlens, and B. Zoph, "Revisiting ResNets: Improved training and scaling strategies," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[13] S. Uhlich, M. Porcu, F. Giron, M. Enenkl, T. Kemp, N. Takahashi, and Y. Mitsufuji, "Improving music source separation based on deep neural networks through data augmentation and network blending," in *Proc. ICASSP*, 2017, pp. 261–265.

[14] A. Défossez, N. Usunier, L. Bottou, and F. Bach, "Music source separation in the waveform domain," *arXiv preprint arXiv:1911.13254*, 2019.

[15] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Proc. Interspeech*, 2015, pp. 3586–3589.

[16] Y. Hu, Y. Liu, S. Lv, M. Xing, S. Zhang, Y. Fu, J. Wu, B. Zhang, and L. Xie, "DCCRN: Deep complex convolution recurrent network for phase-aware speech enhancement," in *Proc. Interspeech*, 2020, pp. 2472–2476.

[17] X. Hao, X. Su, R. Horaud, and X. Li, "FullSubNet: A full-band and sub-band fusion model for real-time single-channel speech enhancement," in *Proc. ICASSP*, 2021, pp. 6633–6637.

[18] M. W. Lam, J. Wang, D. Su, and D. Yu, "Sandglasset: A light multi-granularity self-attentive network for time-domain speech separation," in *Proc. ICASSP*, 2021, pp. 5759–5763.

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

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

[21] C. Li, Y. Luo, C. Han, J. Li, T. Yoshioka, T. Zhou, M. Delcroix, K. Kinoshita, C. Boeddeker, Y. Qian *et al.*, "Dual-path RNN for long recording speech separation," in *Proc. IEEE SLT*, 2021, pp. 865–872.

[22] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "SDR–half-baked or well done?" in *Proc. ICASSP*, 2019, pp. 626–630.

[23] M. Maciejewski, G. Wichern, and J. Le Roux, "WHAMR!: Noisy and reverberant single-channel speech separation," in *Proc. ICASSP*, 2020, pp. 1368–1372.

[24] T. von Neumann, K. Kinoshita, C. Boeddeker, M. Delcroix, and R. Haeb-Umbach, "SA-SDR: A novel loss function for separation of meeting style data," *arXiv preprint arXiv:2110.15581*, 2021.

[25] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong, J.-C. Chou, S.-L. Yeh, S.-W. Fu, C.-F. Liao, E. Rastorgueva, F. Grondin, W. Aris, H. Na, Y. Gao, R. D. Mori, and Y. Bengio, "SpeechBrain: A general-purpose speech toolkit," *arXiv preprint arXiv:2106.04624*, 2021.

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

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

[28] S.-I. Ng and T. Lee, "Data augmentation with locally-time reversed speech for automatic speech recognition," *arXiv preprint arXiv:2110.04511*, 2021.

[29] Y.-Y. Yang, M. Hira, Z. Ni, A. Chourdia, A. Astafurov, C. Chen, C.-F. Yeh, C. Puhrsch, D. Pollack, D. Genzel, D. Greenberg, E. Z. Yang, J. Lian, J. Mahadeokar, J. Hwang, J. Chen, P. Goldsborough, P. Roy, S. Narenthiran, S. Watanabe, S. Chintala, V. Quenneville-Bélair, and Y. Shi, "TorchAudio: Building blocks for audio and speech processing," *arXiv preprint arXiv:2110.15018*, 2021.

[30] J. Benesty, M. M. Sondhi, Y. Huang *et al.*, *Springer handbook of speech processing*. Springer, 2008, vol. 1.

[31] B. Kadıoğlu, M. Horgan, X. Liu, J. Pons, D. Darcy, and V. Kumar, "An empirical study of Conv-TasNet," in *Proc. ICASSP*, 2020, pp. 7264–7268.