

# MixCycle: Unsupervised Speech Separation via Cyclic Mixture Permutation Invariant Training

Ertuğ Karamatlı  and Serap Kırılmaz 

**Abstract**—We introduce two unsupervised source separation methods, which involve self-supervised training from single-channel two-source speech mixtures. Our first method, mixture permutation invariant training (MixPIT), enables learning a neural network model which separates the underlying sources via a challenging proxy task without supervision from the reference sources. Our second method, cyclic mixture permutation invariant training (MixCycle), uses MixPIT as a building block in a cyclic fashion for continuous learning. MixCycle gradually converts the problem from separating mixtures of mixtures into separating single mixtures. We compare our methods to common supervised and unsupervised baselines: permutation invariant training with dynamic mixing (PIT-DM) and mixture invariant training (MixIT). We show that MixCycle outperforms MixIT and reaches a performance level very close to the supervised baseline (PIT-DM) while circumventing the over-separation issue of MixIT. Also, we propose a self-evaluation technique inspired by MixCycle that estimates model performance without utilizing any reference sources. We show that it yields results consistent with an evaluation on reference sources (LibriMix) and also with an informal listening test conducted on a real-life mixtures dataset (REAL-M).

**Index Terms**—Blind source separation, deep learning, self-supervised learning, unsupervised learning.

## I. INTRODUCTION

RECENT state-of-the-art speech separation methods [1], [2], [3], which employ permutation invariant training (PIT) [4], [5], have achieved separation results with almost no perceptible distortion. However, supervised speech separation requires a large dataset of mixture recordings and the corresponding ground truth source recordings, which is challenging and impractical to acquire in the same acoustic environment [6]. Therefore, current methods are usually trained on synthetic mixtures that are generated by mixing clean single-speaker recordings which may not reflect real-life mixture recordings and still require a large dataset of clean recordings.

To avoid this data collection problem, weakly-supervised methods [7], [8], unsupervised methods [9], [10], [11] and

self-supervised representation learning [12] can be employed. Recently, mixture invariant training (MixIT) [13] has been proposed which enables unsupervised training by using single-channel mixtures as references and artificial mixture of mixtures (MoMs) as input. However, MixIT estimates a greater number of sources than the number of underlying sources in the test stage which can cause an over-separation problem where parts of the source signals get spread out between the outputs. In [14], applying sparsity, covariance and classification losses to MixIT is proposed to lessen the over-separation issue, and also a computationally efficient approximation is introduced to handle a larger number of sources. In [15], adapting speech separation to real-world meetings using MixIT is proposed. Teacher-student MixIT [16] addresses the over-separation issue of MixIT by training another model where the number of outputs matches the number of underlying sources.

Mixup-Breakdown [17] is a semi-supervised separation method with a mean-teacher [18] model that improves the generalization capability to mismatch conditions. Although our teacher-student arrangement is similar, our methods are trained from scratch in a purely unsupervised manner and do not require the calculation of moving averages to update the teacher parameters (i.e. simpler implementation, less resource demand). Also, we remix pairs of the teacher source estimates originating from different mixtures instead of remixing pairs from the same mixtures so that our student model is trained on an extremely large number of unique mixtures.

RemixIT [19] is a self-supervised method for speech enhancement. Although our remixing strategy is reminiscent of the bootstrapped remixing approach in [19], our methods are purely unsupervised and do not require supervised pre-training. Also, our methods avoid the over-separation problem by using PIT instead of MixIT.

In this work, we explore training source separation models without having access to the ground truth source signals that constitute single-channel two-source speech mixtures. Our main contributions are summarized as follows:

- 1) We present two purely unsupervised source separation methods that are based on self-supervised [20] training: MixPIT and MixCycle. MixPIT uses a challenging proxy task to avoid the over-separation problem of MixIT [13]. MixCycle uses MixPIT as a building block and applies it in a cyclic fashion for continuous learning.
- 2) On a standard speech separation dataset (LibriMix) [21], we observe that MixCycle attains a performance close to supervised training, which is uncommon in the related works. We also observe that training it on only 5% of the dataset obtains a performance close to training on 100% of the dataset, which demonstrates its data efficiency.

Manuscript received 5 November 2022; revised 21 December 2022; accepted 22 December 2022. Date of publication 26 December 2022; date of current version 6 January 2023. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Daniele Giacobello. (Corresponding author: Ertuğ Karamatlı.)

Ertuğ Karamatlı is with the Department of Computer Engineering, Boğaziçi University, 34342 Istanbul, Turkey (e-mail: ertug@karamatli.com).

Serap Kırılmaz is with the Department of Electrical and Electronics Engineering, MEF University, 34396 Istanbul, Turkey (e-mail: kirbiz@mef.edu.tr).

This letter has supplementary downloadable material available at <https://doi.org/10.1109/LSP.2022.3232276>, provided by the authors.

Digital Object Identifier 10.1109/LSP.2022.3232276

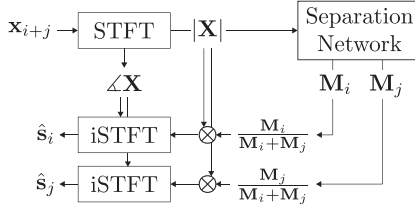


Fig. 1. Our base model  $f_\theta$  with parameters  $\theta$ . The time-domain mixture signal  $\mathbf{x}_{i+j}$  is converted into magnitude  $|\mathbf{X}|$  and phase  $\angle \mathbf{X}$  via short-time Fourier transform (STFT). The separation network is based on the Conv-TasNet [23] architecture, but any other architecture can also be used. The separation network is supplied with the mixture magnitude  $|\mathbf{X}|$ , and it estimates the masks  $M_i$  and  $M_j$ . The masks are ensured to add up to one and multiplied with the mixture magnitude  $|\mathbf{X}|$ . The resulting source magnitude estimates are passed through a differentiable inverse STFT (iSTFT) to arrive at the time-domain source signal estimates  $\hat{\mathbf{s}}_i$  and  $\hat{\mathbf{s}}_j$ .

- 3) We propose a self-evaluation technique inspired by MixCycle, which estimates scale-invariant signal-to-noise ratio improvement (SI-SNRI) [22] without any access to the reference sources. It yields similar results to a ground-truth evaluation on LibriMix and consistent results with an informal listening test we conducted on a real-life mixtures dataset (REAL-M [6]).

## II. BACKGROUND

### A. Permutation Invariant Training (PIT)

We define a supervised training dataset  $\mathcal{X}_s = \{(\mathbf{x}_{i+j}, \mathbf{s}_i, \mathbf{s}_j)\}_{i+j}$  where  $\mathbf{x}_{i+j} = \mathbf{s}_i + \mathbf{s}_j$  is a mixture signal of the time-domain source signals  $\mathbf{s}_i, \mathbf{s}_j \in \mathbb{R}^L$  with length  $L$ . The model  $\hat{\mathbf{S}} = f_\theta(\mathbf{x}_{i+j})$  outputs the source estimates  $\hat{\mathbf{s}}_i, \hat{\mathbf{s}}_j$  in the rows of  $\hat{\mathbf{S}}$ . The details of our base model  $f_\theta$  are given in Fig. 1. The loss function for utterance-level PIT [5] is

$$\mathcal{L}_{\text{PIT}}(\mathbf{s}_i, \mathbf{s}_j, \hat{\mathbf{S}}) = \min_{\mathbf{P}} \left[ \mathcal{L}(\mathbf{s}_i, [\mathbf{P}\hat{\mathbf{S}}]_1) + \mathcal{L}(\mathbf{s}_j, [\mathbf{P}\hat{\mathbf{S}}]_2) \right], \quad (1)$$

where  $\mathbf{P}$  is a  $2 \times 2$  permutation matrix,  $[\cdot]_r$  selects the  $r$ -th row of a matrix and  $\mathcal{L}$  is the loss function calculated between reference sources and their estimates, as illustrated in Fig. 2(a).

### B. Mixture Invariant Training (MixIT)

We define an unsupervised training dataset  $\mathcal{X}_u = \{(\mathbf{x}_{i+j}, \mathbf{x}_{k+l})\}_{(i+j, k+l)}$ . The input to the model is formed by summing two mixtures  $\mathbf{x}_{i+j}$  and  $\mathbf{x}_{k+l}$ . The model  $\hat{\mathbf{S}} = f_\theta(\mathbf{x}_{i+j} + \mathbf{x}_{k+l})$  outputs the source estimates  $\hat{\mathbf{s}}_i, \hat{\mathbf{s}}_j, \hat{\mathbf{s}}_k, \hat{\mathbf{s}}_l$ . The loss function for MixIT [13] is calculated between reference mixtures and their estimates as

$$\mathcal{L}_{\text{MixIT}}(\cdot) = \min_{\mathbf{A}} \left[ \mathcal{L}(\mathbf{x}_{i+j}, [\mathbf{A}\hat{\mathbf{S}}]_1) + \mathcal{L}(\mathbf{x}_{k+l}, [\mathbf{A}\hat{\mathbf{S}}]_2) \right] \quad (2)$$

where  $\mathbf{A}$  is a  $2 \times 4$  binary mixing matrix, as illustrated in Fig. 2(b). In the test stage, a single mixture is supplied to the model  $f_\theta(\mathbf{x}_{i+j})$ .

## III. PROPOSED METHODS

### A. Mixture Permutation Invariant Training (MixPIT)

The main limitation of MixIT is the over-separation problem which stems from having a greater number of model outputs than the actual number of underlying sources. Here, we remove

TABLE I  
ALL OF THE POSSIBLE CASES FOR THE MIXPIT LOSS FUNCTION

Model output pair	Perm.	Loss function $\mathcal{L}_{\text{MixPIT}}(\cdot)$	MSI
$\hat{\mathbf{S}}_1 = \{\hat{\mathbf{x}}_{i+j}, \hat{\mathbf{x}}_{k+l}\}$	1	$\mathcal{L}(\mathbf{x}_{i+j}, \hat{\mathbf{x}}_{i+j}) + \mathcal{L}(\mathbf{x}_{k+l}, \hat{\mathbf{x}}_{k+l})$	$i, j, k, l$
	2	$\mathcal{L}(\mathbf{x}_{i+j}, \hat{\mathbf{x}}_{k+l}) + \mathcal{L}(\mathbf{x}_{k+l}, \hat{\mathbf{x}}_{i+j})$	-
$\hat{\mathbf{S}}_2 = \{\hat{\mathbf{x}}_{i+k}, \hat{\mathbf{x}}_{j+l}\}$	1	$\mathcal{L}(\mathbf{x}_{i+j}, \hat{\mathbf{x}}_{i+k}) + \mathcal{L}(\mathbf{x}_{k+l}, \hat{\mathbf{x}}_{j+l})$	$i, l$
	2	$\mathcal{L}(\mathbf{x}_{i+j}, \hat{\mathbf{x}}_{j+l}) + \mathcal{L}(\mathbf{x}_{k+l}, \hat{\mathbf{x}}_{i+k})$	$j, k$
$\hat{\mathbf{S}}_3 = \{\hat{\mathbf{x}}_{j+k}, \hat{\mathbf{x}}_{i+l}\}$	1	$\mathcal{L}(\mathbf{x}_{i+j}, \hat{\mathbf{x}}_{j+k}) + \mathcal{L}(\mathbf{x}_{k+l}, \hat{\mathbf{x}}_{i+l})$	$j, l$
	2	$\mathcal{L}(\mathbf{x}_{i+j}, \hat{\mathbf{x}}_{i+l}) + \mathcal{L}(\mathbf{x}_{k+l}, \hat{\mathbf{x}}_{j+k})$	$i, k$

this limitation by training a model where we have four sources (i.e. a mixture of mixtures) to separate but only have two model outputs. We use the model  $f_\theta(\mathbf{x}_{i+j} + \mathbf{x}_{k+l})$  which has three possible output pairs  $\hat{\mathbf{S}}_1, \hat{\mathbf{S}}_2, \hat{\mathbf{S}}_3$  as given in Table I, ignoring the output permutations (abbreviated to ‘‘Perm.’’). The case of the output pair  $\hat{\mathbf{S}}_1$  is illustrated in Fig. 2(c).

Assuming that the sources  $\mathbf{s}_i, \mathbf{s}_j, \mathbf{s}_k, \mathbf{s}_l$  are statistically independent from each other, the output pairs  $\hat{\mathbf{S}}_1, \hat{\mathbf{S}}_2, \hat{\mathbf{S}}_3$  are equally likely because the model cannot learn the difference between the possible source pairings of the input mixtures:  $\mathbf{x}_{i+j} + \mathbf{x}_{k+l}, \mathbf{x}_{i+k} + \mathbf{x}_{j+l}$  and  $\mathbf{x}_{j+k} + \mathbf{x}_{i+l}$ .

We train the model  $f_\theta$  using the loss function

$$\mathcal{L}_{\text{MixPIT}}(\cdot) = \mathcal{L}_{\text{PIT}}(\mathbf{x}_{i+j}, \mathbf{x}_{k+l}, \hat{\mathbf{S}}), \quad (3)$$

where  $\hat{\mathbf{S}} \in \{\hat{\mathbf{S}}_1, \hat{\mathbf{S}}_2, \hat{\mathbf{S}}_3\}$  is the model output pair. Table I lists all of the possible instances of  $\mathcal{L}_{\text{MixPIT}}(\cdot)$ . For each of these instances, matching source indices (MSI) are listed as well.

The PIT loss  $\mathcal{L}_{\text{PIT}}(\cdot)$  in (1) selects the best output permutation that minimizes the loss value. Therefore, the output pair  $\hat{\mathbf{S}}_1$  corresponds to an *exact match* where all of the sources match (Perm. 1 is always selected due to having the lower loss value), while  $\hat{\mathbf{S}}_2$  and  $\hat{\mathbf{S}}_3$  correspond to a *partial match* where two of the sources match (Perm. 1 and 2 are equally likely). Due to the fact that at least two sources are guaranteed to match in the loss function, the model learns to separate two sources as well, despite some noise from mismatching sources.

Ultimately, the model is trained with the challenging proxy task of separating mixtures of mixtures which also covers our main objective of separating single mixtures. In the test stage, we supply the model with a single mixture  $f_\theta(\mathbf{x}_{i+j})$  to obtain the source estimates  $\hat{\mathbf{s}}_i$  and  $\hat{\mathbf{s}}_j$ . There is no over-separation due to the equal number of model outputs and underlying sources.

### B. Cyclic Mixture Permutation Invariant Training (MixCycle)

Here we propose a new method that improves the performance further by using MixPIT as a building block and applying it in a cyclic fashion for continuous learning.

First, we use a teacher model  $f_{\theta'}$  to estimate four sources from the two input mixtures

$$\{\tilde{\mathbf{s}}_i, \tilde{\mathbf{s}}_j\} = f_{\theta'}(\mathbf{x}_{i+j}), \quad \{\tilde{\mathbf{s}}_k, \tilde{\mathbf{s}}_l\} = f_{\theta'}(\mathbf{x}_{k+l}) \quad (4)$$

where  $\theta' = \theta^{(\tau-1)}$  are the parameters at the previous training step  $\tau - 1$ . Second, we use these estimated sources to generate unique mixtures such that each constituent source estimate originates from a different randomly-selected mixture. We accomplish this by randomly choosing one of the following two remixing options, which eliminates any bias due to model output permutations:

$$\text{Opt. 1: } \tilde{\mathbf{x}}_{i+k} = \tilde{\mathbf{s}}_i + \tilde{\mathbf{s}}_k, \quad \tilde{\mathbf{x}}_{j+l} = \tilde{\mathbf{s}}_j + \tilde{\mathbf{s}}_l \quad (5)$$

$$\text{Opt. 2: } \tilde{\mathbf{x}}_{j+k} = \tilde{\mathbf{s}}_j + \tilde{\mathbf{s}}_k, \quad \tilde{\mathbf{x}}_{i+l} = \tilde{\mathbf{s}}_i + \tilde{\mathbf{s}}_l. \quad (6)$$

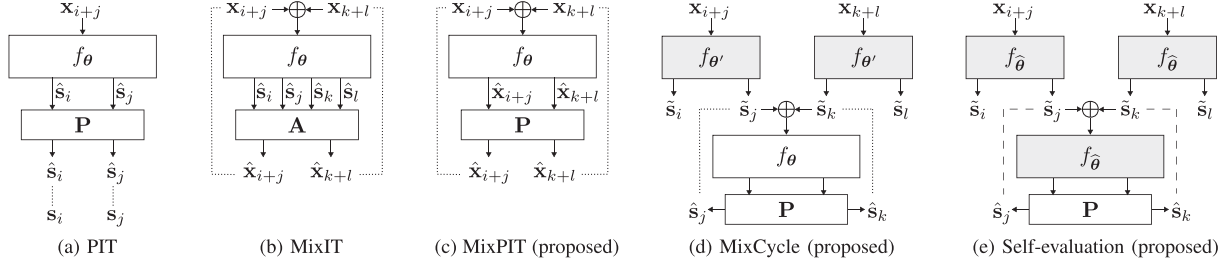


Fig. 2. Illustration of the methods that we consider in this work. The dotted lines denote the calculation of the loss function  $\mathcal{L}$  between various signals. (a) In PIT, the mixture  $\mathbf{x}_{i+j}$  is inputted to the model  $f_\theta$ . The best match between the two model outputs and the sources  $\mathbf{s}_i, \mathbf{s}_j$  is chosen by a permutation matrix  $\mathbf{P}$  to update the parameters  $\theta$ . (b) In MixIT, the mixtures  $\mathbf{x}_{i+j}, \mathbf{x}_{k+l}$  are summed and inputted to the model  $f_\theta$ . The best match between the four model outputs and the mixtures  $\mathbf{x}_{i+j}, \mathbf{x}_{k+l}$  is chosen by a mixing matrix  $\mathbf{A}$  to update the parameters  $\theta$ . (c) In MixPIT, the mixtures  $\mathbf{x}_{i+j}, \mathbf{x}_{k+l}$  are summed and inputted to the model  $f_\theta$ . The best match between the two model outputs and the mixtures  $\mathbf{x}_{i+j}, \mathbf{x}_{k+l}$  is chosen by a permutation matrix  $\mathbf{P}$  to update the parameters  $\theta$ . (d) In MixCycle, the mixtures  $\mathbf{x}_{i+j}, \mathbf{x}_{k+l}$  are inputted to the teacher model  $f_{\theta'}$ . The teacher outputs are paired such that each pair contains source estimates from two different original mixtures:  $\{\tilde{\mathbf{s}}_j, \tilde{\mathbf{s}}_k\}, \{\tilde{\mathbf{s}}_i, \tilde{\mathbf{s}}_l\}$ . Each pair is summed to produce an artificial mixture before being inputted to the student model  $f_\theta$ . Considering the pair  $\{\tilde{\mathbf{s}}_j, \tilde{\mathbf{s}}_k\}$ , the best match between the two student outputs and the teacher source estimates  $\tilde{\mathbf{s}}_j, \tilde{\mathbf{s}}_k$  is chosen by a permutation matrix  $\mathbf{P}$  to update the parameters  $\theta$  while  $\theta'$  are frozen. The other pair  $\{\tilde{\mathbf{s}}_i, \tilde{\mathbf{s}}_l\}$  is also processed similarly. (e) In self-evaluation, given a trained model  $f_{\hat{\theta}}$ , the ground-truth reference sources are estimated as  $\tilde{\mathbf{s}}_i, \tilde{\mathbf{s}}_j, \tilde{\mathbf{s}}_k, \tilde{\mathbf{s}}_l$ . These estimated reference sources are paired and summed as in MixCycle. The trained model  $f_{\hat{\theta}}$  is evaluated on the resulting artificial mixtures where the dashed lines denote the calculation of SI-SNRi between the source estimates  $\hat{\mathbf{s}}_j, \hat{\mathbf{s}}_k$  and the estimated reference sources  $\tilde{\mathbf{s}}_j, \tilde{\mathbf{s}}_k$ . (a) PIT (b) MixIT (c) MixPIT (proposed) (d) MixCycle (proposed) (e) Self-evaluation (proposed).

Finally, we train a student model  $f_\theta$  on these artificial mixtures with the loss function (no backpropagation into  $f_{\theta'}$ )

$$\mathcal{L}_{\text{MixCycle}}(\cdot) = \mathcal{L}_{\text{PIT}}(\tilde{\mathbf{s}}_j, \tilde{\mathbf{s}}_k, \hat{\mathbf{S}}_1) + \mathcal{L}_{\text{PIT}}(\tilde{\mathbf{s}}_i, \tilde{\mathbf{s}}_l, \hat{\mathbf{S}}_2) \quad (7)$$

to estimate the sources (assuming Opt. 2 is chosen):

$$\hat{\mathbf{S}}_1 = \{\hat{\mathbf{s}}_j, \hat{\mathbf{s}}_k\} = f_\theta(\tilde{\mathbf{x}}_{j+k}), \quad \hat{\mathbf{S}}_2 = \{\hat{\mathbf{s}}_i, \hat{\mathbf{s}}_l\} = f_\theta(\tilde{\mathbf{x}}_{i+l}) \quad (8)$$

where  $\theta = \theta^{(\tau)}$  are the parameters at the current training step  $\tau$ . The model is illustrated in Fig. 2(d).

We designed the model  $f_\theta$  such that it produces informative initial source estimates and helps prevent the source estimates from diverging throughout the training process. We accomplish this by employing time-frequency masking and ensuring that the masks add up to one as given in Fig. 1. Therefore, we have mixture consistency [24] as  $\tilde{\mathbf{s}}_i + \tilde{\mathbf{s}}_j = \mathbf{x}_{i+j}$ . Also, our remixing strategy acts as a data augmentation mechanism by generating an extremely large number of unique mixtures on-the-fly. Consequently, it increases the effective size of the available training set, similar to dynamic mixing [25].

MixCycle can be viewed as a cascade of successive MixPIT training steps with a continuously improved mixture input instead of a static mixture of mixtures input. To elaborate on this, we define the initial source estimates  $\tilde{\mathbf{s}}_j^{(0)}, \tilde{\mathbf{s}}_k^{(0)}$  of the teacher model  $f_{\theta'}$  in (4) when the parameters  $\theta' = \theta^{(0)}$  are randomly initialized at the first training step  $\tau = 1$ :

$$\tilde{\mathbf{s}}_j^{(0)} = \text{iSTFT} \left( \left| \mathbf{X}_{i+j} \right| \frac{\mathbf{M}_j^{(0)}}{\mathbf{M}_i^{(0)} + \mathbf{M}_j^{(0)}}, \angle \mathbf{X}_{i+j} \right) \quad (9)$$

$$\tilde{\mathbf{s}}_k^{(0)} = \text{iSTFT} \left( \left| \mathbf{X}_{k+l} \right| \frac{\mathbf{M}_k^{(0)}}{\mathbf{M}_k^{(0)} + \mathbf{M}_l^{(0)}}, \angle \mathbf{X}_{k+l} \right) \quad (10)$$

where  $|\mathbf{X}_{i+j}|, |\mathbf{X}_{k+l}|$  and  $\angle \mathbf{X}_{i+j}, \angle \mathbf{X}_{k+l}$  are the magnitude and phase spectrograms of the mixture signals  $\mathbf{x}_{i+j}, \mathbf{x}_{k+l}$ , respectively.  $\mathbf{M}_i^{(0)}, \mathbf{M}_j^{(0)}, \mathbf{M}_k^{(0)}, \mathbf{M}_l^{(0)} \in (0, 1)^{F \times T}$  are noisy mask outputs of the randomly initialized model with  $F$  frequency bins and  $T$  time frames. If we consider the initial input mixture  $\tilde{\mathbf{x}}_{j+k}^{(0)} = \tilde{\mathbf{s}}_j^{(0)} + \tilde{\mathbf{s}}_k^{(0)}$  of the student model  $f_\theta(\cdot)$  in (8) using (9) and (10), we can see that this is similar to the proposed

MixPIT method because we have a mixture of noisy mixtures  $\tilde{\mathbf{x}}_{j+k}^{(0)}$  as input and try to separate it into single noisy mixtures  $\tilde{\mathbf{s}}_j^{(0)}, \tilde{\mathbf{s}}_k^{(0)}$ . In contrast to MixPIT, the input mixture  $\tilde{\mathbf{x}}_{j+k}$  is not static and refined at each training step  $\tau$  such that the constituent teacher source estimates  $\tilde{\mathbf{s}}_j, \tilde{\mathbf{s}}_k$ , which start as noisy copies of the original mixtures, are transformed into accurate estimates of the corresponding sources  $\mathbf{s}_j, \mathbf{s}_k$  as we optimize the parameters  $\theta$ .

In practice, the initial source estimates are very noisy due to the random initialization. To reduce the noise and stabilize the training process, we initialize the model by training it with the proposed MixPIT method for the first  $I$  epochs.

#### IV. EXPERIMENTS

We evaluate the proposed methods on two two-speaker datasets: a standard speech separation dataset (LibriMix [21]) and a recently-released real-life mixtures dataset (REAL-M [6]) which has no ground-truth reference sources. For LibriMix, we used the clean version of the train-360 split with its *min* mode and an 8 kHz sampling rate. The training, validation and test sets contain 212, 11 and 11 hours of speech mixtures, respectively. We refer to the complete training set as the *100% dataset* and its 5% random subset as the *5% dataset* while keeping the original validation and test sets. For REAL-M, we discarded the ‘‘early collection’’ subset, which showed higher variation in difficulty, and split the remaining data into training and validation sets with 61 and 13 minutes of mixed speech, respectively.

As given in [23], the best performing Conv-TasNet configuration uses  $D = 8$  dilated convolutions in each repeated block. It also uses a window size of 16 and a hop size of 8 for its learned representation. On the other hand, we use STFT/iSTFT with a window size of 512, a hop size of 128, and a Hann window. To compensate for our shorter representation, we used  $D = 4$  which keeps the receptive field of the stacked dilated convolutions similar between the representations.

We randomly sampled three-second-long segments from utterances while training. We used the negative thresholded signal-to-noise ratio with  $\text{SNR}_{\text{max}} = 30$  dB as in [13] for the loss function  $\mathcal{L}$ . We used the Adam optimization algorithm with its default

TABLE II  
EVALUATION ON THE LIBRIMIX TEST SET IN TERMS OF SI-SNRi

Method	Super.	5% (dB)	100% (dB)	Step (s)	Total (h)
Conv-TasNet [21]	Yes	–	14.7	–	–
IRM	Oracle	–	13.9±2.5	–	–
PIT	Yes	7.1±5.0	11.2±3.7	0.26	11.0
PIT-DM	Yes	11.4±3.4	11.8±3.2	0.26	11.9
MixIT (oracle)	No	8.5±3.8	9.9±3.4	0.46	27.6
MixIT	No	6.0±3.5	7.8±3.6	0.46	27.6
MixPIT (proposed)	No	5.6±4.0	7.1±3.8	0.28	4.2
MixCycle (proposed)	No	<b>11.2±3.5</b>	<b>11.4±3.3</b>	0.32	17.9

parameters [26] and a batch size of 128. We applied gradient clipping with a maximum  $L_2$ -norm of 5, which was calculated over all gradients together. We employed early stopping and used the models with the best validation scores for testing. We initialized the MixCycle models with MixPIT for 50 and 250 epochs on the 100% and 5% datasets, respectively. We used LibriMix to initialize the MixCycle model for REAL-M because the REAL-M dataset was too small ( $\sim 1$  hour of speech mixtures) for this task.

We evaluate performance using SI-SNRi [22]. We find the best match between the reference sources and the model outputs to calculate the SI-SNRi. We used PyTorch with an NVIDIA GTX 1080 Ti GPU to develop and evaluate our methods. We released the source code<sup>1</sup> for reproducibility and published audio samples<sup>1</sup> to demonstrate our results.

#### A. Results on LibriMix

The proposed MixPIT and MixCycle methods are compared with supervised and unsupervised baselines: Conv-TasNet [23], ideal ratio mask (IRM) [27], PIT [5], PIT with dynamic mixing [25] (PIT-DM), and unsupervised MixIT [13]. Table II shows the performance of the methods trained on the 5% and 100% datasets (LibriMix) in terms of SI-SNRi, reporting means and standard deviations. We also report the time per training step (Step) in seconds (s) and total training time (Total) in hours (h) on the 100% dataset. The supervised methods (Super.) can be considered as an empirical upper bound on the performance of the unsupervised methods. All of the baseline methods are our implementation except Conv-TasNet. We observed that the performance of MixPIT is only slightly behind MixIT, despite its challenging training objective and lower computational demand. We also include an oracle evaluation of the same MixIT model by using the mixing matrix  $\mathbf{A}$  to find the best match between the remixed outputs and reference sources as in [16]. The gap between the oracle and standard evaluation demonstrates the significance of the over-separation issue.

MixCycle improved upon MixPIT as expected and achieved the best performance among the unsupervised methods. Also, it reached a performance level that is close to supervised training (PIT-DM). This is due to the teacher model estimating the sources more accurately as the training progresses; therefore, the student model gets trained on almost the same supervised dataset with dynamic mixing that PIT-DM has access to. We can see the data efficiency of MixCycle on the 5% dataset as the performance approached that on the 100% dataset.

#### B. Results on REAL-M

The results on LibriMix in the previous section has shown that a teacher model combined with our remixing strategy produces

<sup>1</sup><https://github.com/ertug/MixCycle>

TABLE III  
SELF-EVALUATION ON THE VALIDATION SETS IN TERMS OF SI-SNRi

Training Setup		LibriMix		REAL-M	
Method	Dataset	GE (dB)	SE (dB)	SE (dB)	MOS
PIT-DM	LibriMix	12.0±3.4	12.0±3.9	9.6±5.5	2.9±1.2
MixPIT	LibriMix	7.7±3.9	7.0±3.6	5.5±3.8	–
MixCycle	LibriMix	11.6±3.6	11.9±4.0	10.0±5.6	–
MixCycle	REAL-M	10.0±4.4	10.0±4.7	<b>13.3±4.5</b>	<b>3.4±1.2</b>

such accurate artificial mixtures that there is not much difference between training on the artificial (i.e. MixCycle) and original (i.e. PIT-DM) mixtures. Therefore, we propose repurposing MixCycle as a self-evaluation technique to estimate SI-SNRi as illustrated in Fig. 2(e). First, given a trained model  $f_{\hat{\theta}}$ , we utilize it as  $f_{\hat{\theta}}(\mathbf{x}_{i+j}), f_{\hat{\theta}}(\mathbf{x}_{k+l})$  to estimate the missing ground-truth reference sources as  $\tilde{\mathbf{s}}_i, \tilde{\mathbf{s}}_j, \tilde{\mathbf{s}}_k, \tilde{\mathbf{s}}_l$ . Second, we apply the remixing strategy to generate unique mixtures such as  $\tilde{\mathbf{x}}_{j+k} = \tilde{\mathbf{s}}_j + \tilde{\mathbf{s}}_k$ . Finally, we evaluate the trained model  $f_{\hat{\theta}}$  on these artificial mixtures  $\tilde{\mathbf{x}}_{j+k}$ . We repeat this procedure 100 times, which creates additional unique mixtures and uses them for evaluation. This can be viewed as applying dynamic mixing to a noisy version of the validation set, thus increasing its effective size and improving the reliability of the results.

Table III compares different training setups according to ground-truth evaluation (GE), self-evaluation (SE) and mean opinion scores (MOS) on the validation sets of LibriMix and REAL-M, reporting means and standard deviations. We omit MixIT because it is not apparent how to select the correct two model outputs (out of four) when the reference sources are not available. The ground-truth evaluation and self-evaluation results on LibriMix are similar as expected. Therefore, we use self-evaluation (SE) as a surrogate for ground-truth evaluation (GE) on REAL-M and observe a considerable improvement (+3.7 dB) with MixCycle training on REAL-M over PIT-DM training on LibriMix.

As ground-truth evaluation is impossible on REAL-M, an informal listening test was conducted with 10 participants on 10 randomly-picked REAL-M validation mixtures to back up the self-evaluation results. The participants (4 female and 6 male), aged between 21 and 41 years, rated the separation results online while using headphones. The participants were asked to give an overall quality score between 1-5 (higher is better), considering both the sound quality of the target source and the interference from the other source. We obtained mean opinion scores (MOS) as given in Table III, which are consistent with the corresponding self-evaluation (SE) results. We also provide the audio samples<sup>1</sup> used in this test.

## V. CONCLUSION

We introduced unsupervised speech separation methods that avoid over-separation and narrow the performance gap between supervised and unsupervised training. We defer exploring mixtures with more than two sources and different source classes (these can pose a greater challenge in training the models) to future work. Also, we proposed a promising self-evaluation technique that we intend to investigate further.

## ACKNOWLEDGMENT

The authors would like to thank Ali Taylan Cemgil, Cem Subakan and the anonymous reviewers for their insightful comments.

## REFERENCES

- [1] C. Subakan, M. Ravanelli, S. Cornell, M. Bronzi, and J. Zhong, "Attention is all you need in speech separation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2021, pp. 21–25.
- [2] E. Nachmani, Y. Adi, and L. Wolf, "Voice separation with an unknown number of multiple speakers," in *Proc. 37th Int. Conf. Mach. Learn.*, 2020, pp. 7164–7175.
- [3] E. Tzinis, Z. Wang, and P. Smaragdis, "Sudo RM-RF: Efficient networks for universal audio source separation," in *Proc. IEEE 30th Int. Workshop Mach. Learn. Signal Process.*, 2020, pp. 1–6.
- [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. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 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 Trans. Audio, Speech, Lang. Process.*, vol. 25, no. 10, pp. 1901–1913, Oct. 2017.
- [6] C. Subakan, M. Ravanelli, S. Cornell, and F. Grondin, "Real-M: Towards speech separation on real mixtures," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2022, pp. 6862–6866.
- [7] E. Karamatli, A. T. Cengil, and S. Kirbiz, "Audio source separation using variational autoencoders and weak class supervision," *IEEE Signal Process. Lett.*, vol. 26, no. 9, pp. 1349–1353, Sep. 2019.
- [8] F. Pishdadian, G. Wichern, and J. L. Roux, "Finding strength in weakness: Learning to separate sounds with weak supervision," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 2386–2399, 2020.
- [9] L. Drude, D. Hasenklever, and R. Haeb-Umbach, "Unsupervised training of a deep clustering model for multichannel blind source separation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 695–699.
- [10] Y. Bando, K. Sekiguchi, Y. Masuyama, A. A. Nugraha, M. Fontaine, and K. Yoshii, "Neural full-rank spatial covariance analysis for blind source separation," *IEEE Signal Process. Lett.*, vol. 28, pp. 1670–1674, 2021.
- [11] J. Neri, R. Badeau, and P. Depalle, "Unsupervised blind source separation with variational auto-encoders," in *Proc. 29th Eur. Signal Process. Conf.*, 2021, pp. 311–315.
- [12] Z. Huang, S. Watanabe, S.-w. Yang, P. García, and S. Khudanpur, "Investigating self-supervised learning for speech enhancement and separation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2022, pp. 6837–6841.
- [13] S. Wisdom, E. Tzinis, H. Erdogan, R. Weiss, K. Wilson, and J. Hershey, "Unsupervised sound separation using mixture invariant training," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 3846–3857.
- [14] S. Wisdom, A. Jansen, R. J. Weiss, H. Erdogan, and J. R. Hershey, "Sparse, efficient, and semantic mixture invariant training: Taming in-the-wild unsupervised sound separation," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust.*, 2021, pp. 51–55.
- [15] A. Sivaraman, S. Wisdom, H. Erdogan, and J. R. Hershey, "Adapting speech separation to real-world meetings using mixture invariant training," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2022, pp. 686–690.
- [16] J. Zhang, C. Zorilă, R. Doddipatla, and J. Barker, "Teacher-student MixIT for unsupervised and semi-supervised speech separation," in *Proc. Interspeech*, 2021, pp. 3495–3499.
- [17] M. W. Lam, J. Wang, D. Su, and D. Yu, "Mixup-breakdown: A consistency training method for improving generalization of speech separation models," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2020, pp. 6374–6378.
- [18] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, vol. 30, pp. 1195–1204.
- [19] E. Tzinis, Y. Adi, V. K. Ithapu, B. Xu, and A. Kumar, "Continual self-training training with bootstrapped remixing for speech enhancement," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2022, pp. 6947–6951.
- [20] L. Weng, "Self-supervised representation learning" 2019. [Online]. Available: <https://lilianweng.github.io/lil-log/2019/11/10/self-supervised-learning.html>
- [21] J. Cosentino, M. Pariente, S. Cornell, A. Deleforge, and E. Vincent, "LibriMix: An open-source dataset for generalizable speech separation," 2020, *arXiv:2005.11262*.
- [22] J. L. Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "SDR-half-baked or well done?," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 626–630.
- [23] Y. Luo and N. Mesgarani, "Conv-TasNet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 8, pp. 1256–1266, Aug. 2019.
- [24] S. Wisdom et al., "Differentiable consistency constraints for improved deep speech enhancement," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 900–904.
- [25] N. Zeghidour and D. Grangier, "WaveSplit: End-to-end speech separation by speaker clustering," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 2840–2849, 2021.
- [26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015.
- [27] Y. Wang, A. Narayanan, and D. Wang, "On training targets for supervised speech separation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 22, no. 12, pp. 1849–1858, Dec. 2014.