RemixIT: Continual self-training of speech enhancement models via bootstrapped remixing

Efthymios Tzinis *Graduate Student Member, IEEE*, Yossi Adi *Member, IEEE*, Vamsi K. Ithapu *Member, IEEE*, Buye Xu *Member, IEEE*, Paris Smaragdis *Fellow, IEEE*, Anurag Kumar *Member, IEEE*

***Abstract*—We present *RemixIT*, a simple yet effective self-supervised method for training speech enhancement without the need of a single isolated in-domain speech nor a noise waveform. Our approach overcomes limitations of previous methods which make them dependent on clean in-domain target signals and thus, sensitive to any domain mismatch between train and test samples. *RemixIT* is based on a continuous self-training scheme in which a pre-trained teacher model on out-of-domain data infers estimated pseudo-target signals for in-domain mixtures. Then, by permuting the estimated clean and noise signals and *remixing* them together, we generate a new set of bootstrapped mixtures and corresponding pseudo-targets which are used to train the student network. Vice-versa, the teacher periodically refines its estimates using the updated parameters of the latest student models. Experimental results on multiple speech enhancement datasets and tasks not only show the superiority of our method over prior approaches but also showcase that *RemixIT* can be combined with any separation model as well as be applied towards any semi-supervised and unsupervised domain adaptation task. Our analysis, paired with empirical evidence, sheds light on the inside functioning of our self-training scheme wherein the student model keeps obtaining better performance while observing severely degraded pseudo-targets.**

***Index Terms*—Self-supervised learning, speech enhancement, semi-supervised self-training, zero-shot domain adaptation.**

I. Introduction

One of the most fundamental problems in audio processing is speech enhancement, where the goal is to isolate and reconstruct the clean speech component from a noisy input recording [1]. Several studies have shown that employing such denoising models as front-ends could be useful for building robust automatic speech recognition (ASR) [2], [3] and speaker recognition [4] systems. The universal applicability of neural networks has proven to be beneficial for a variety of signal processing problems, including speech enhancement. Sophisticated architectures such as convolutional networks [5]–[8], recurrent processing [9] self-attention [10]–[12], generative adversarial networks [13], [14] as well as variational autoencoders [15], to name a few. Despite the effectiveness of the aforementioned approaches in cases where large amounts of in-domain training paired data are available, real-world applications necessitate the need for developing robust algorithms to train these models with in-the-wild mixtures.

In the context of speech enhancement, self-supervised learning (SSL) or unsupervised methods differ from semi-supervised ones [16] in the sense that the former do not have access to clean target signals. Orthogonal to these concepts, self-training refers to algorithms which are able to train a new model (student) based on pseudo-targets provided by a previously fitted model (teacher). Under this unified terminology, the proposed *RemixIT* framework can also be viewed as an unsupervised self-training algorithm when only unsupervised data are used to pre-train the teacher model.

Recent studies have shown that speech representations could be self-learned and be used later for other downstream audio processing tasks [17]–[19]. However, in real-world settings, the speech recordings are degraded with additive noise, thus, self-learning robust embeddings becomes particularly challenging and demands the adaptation to the input noise distribution [20]. Several unsupervised speech denoising algorithms have been proposed by identifying and training with relatively clean segments of the noisy speech mixture [21], [22], using ASR losses [23], [24] exploiting visual cues [25], and harnessing the spatial separability of the sources using mic-arrays [26], [27]. Mixture invariant training (MixIT) [28] enables unsupervised training of separation models only with real-world single-channel recordings by generating artificial mixtures of mixtures and estimating the independent sources. Although MixIT has been proven successful for various speech enhancement tasks [28]–[30], MixIT assumes access to in-domain noise samples which restricts its universal applicability. Overcoming the latter constraint by injecting additional out-of-domain (OOD) noise sources to the input mixture of mixtures [31] further alters the input signal-to-noise ratio (SNR) distribution and its performance depends heavily on the distribution shift between the injected and real noise distributions. Thus, developing a SSL algorithm which does not depend on external modality information nor assumptions about in-domain data remains a challenging problem.

On the other hand, several self-training strategies have emerged and showed promising results in classification tasks using convex combinations of labeled and unlabeled data (e.g. Mixup [32]) but have also been successfully applied to several audio tasks [33], [34]. In [35], a student model with a smaller number of estimated sources has been trained on

E. Tzinis is with the University of Illinois at Urbana-Champaign, Urbana, Illinois, USA (email: etzinis2@illinois.edu). Part of the work was done while E. Tzinis was an intern at Meta Reality Labs Research. Y. Adi is with the Hebrew University of Jerusalem and Meta AI Research, Tel Aviv, Israel (email: adiyoss@fb.com). V. Ithapu and B.Xu are with Meta Reality Labs Research, Redmond, WA, USA (email: ithapu, xub@fb.com). P. Smaragdis is with the University of Illinois at Urbana-Champaign, Urbana, Illinois, USA (email: paris@illinois.edu). A. Kumar is with Meta Reality Labs Research, Redmond, WA, USA (email: anuragkr@fb.com).

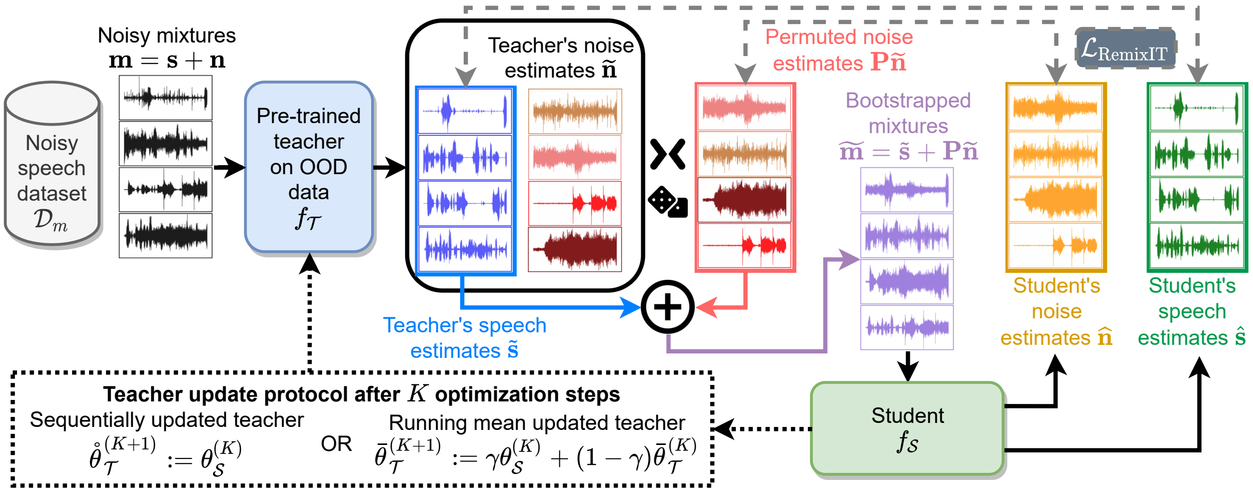

Noisy mixtures $\mathbf{m} = \mathbf{s} + \mathbf{n}$
Noisy speech dataset $\mathcal{D}_m$
Pre-trained teacher on OOD data $f_{\mathcal{T}}$
Teacher's noise estimates $\widetilde{\mathbf{n}}$
Permuted noise estimates $\mathbf{P}\widetilde{\mathbf{n}}$
$\mathcal{L}_{\text{RemixIT}}$
Bootstrapped mixtures $\widetilde{\mathbf{m}} = \widetilde{\mathbf{s}} + \mathbf{P}\widetilde{\mathbf{n}}$
Student's noise estimates $\widehat{\mathbf{n}}$
Student's speech estimates $\widehat{\mathbf{s}}$
Teacher's speech estimates $\widetilde{\mathbf{s}}$
Student $f_{\mathcal{S}}$
Teacher update protocol after K optimization steps
Sequentially updated teacher
$\mathring{\theta}_{\mathcal{T}}^{(K+1)} := \theta_{\mathcal{S}}^{(K)}$
OR
Running mean updated teacher
$\bar{\theta}_{\mathcal{T}}^{(K+1)} := \gamma\theta_{\mathcal{S}}^{(K)} + (1-\gamma)\bar{\theta}_{\mathcal{T}}^{(K)}$

Fig. 1: *RemixIT* self-training procedure with a batch size of 4 noisy mixtures. A teacher speech enhancement model $f_{\mathcal{T}}$ is pre-trained in a supervised or unsupervised way on out-of-domain (OOD) data and performs inference on a batch of noisy mixtures sampled from the in-domain noisy speech dataset $\mathbf{m} \sim \mathcal{D}_m$. The randomly permuted teacher's noise estimates $\mathbf{P}\widetilde{\mathbf{n}}$ are added together with the teacher's speech estimates $\widetilde{\mathbf{s}}$ to form the bootstrapped mixtures $\widetilde{\mathbf{m}}$ which are fed to the student speech enhancement network $f_{\mathcal{S}}$. The student is trained by regressing over the teacher's estimated sources which are now used as pseudo-targets under a specified signal-level loss function. After repeating the overall process for K optimization steps, the teacher model may be updated using the student's weights in a continuous self-training scheme.

a subset of outputs of a pre-trained MixIT model to solve the input SNR distribution mismatch. Furthermore, a student model could also perform test-time adaptation by using the teacher's estimated waveforms as targets [36]. However, those approaches enforce only the consistency of the student's predictions over a frozen teacher's output pseudo-targets whereas other studies have shown that one can obtain significant gains using unsupervised data augmentation [37], averages of losses over multiple predictions [38], or their combination [39].

The student-teacher framework for singing-voice separation in [40] bears the closest similarity to our work. The proposed setup assumes teacher pre-training on supervised OOD data, performing inference on the in-domain noisy dataset and storing the new pseudo-labeled dataset. At a second step, a student network is trained on randomly mixed estimated sources that score above a pre-defined confidence quality threshold. Unfortunately, if the teacher's estimates have low SNR and/or the threshold is not picked wisely then the student model would also perform poorly. In contrast, some of the most successful self-training approaches propose to iteratively update the teacher's weights using an exponential moving average scheme [41]–[43] or sequentially update the teacher with the weights from a more expressive noisy student [44].

In this work, we propose *RemixIT* which is based on several aforementioned state-of-the-art SSL strategies for pseudo-labeling and continual training while also providing a novel technique for training speech enhancement models with OOD data. Our method trains a student model using self-augmented mixtures generated by permuting and remixing the teacher's estimates and using them as pseudo-targets for regular regression. Moreover, *RemixIT* treats self-training as a lifelong process while continually updating the teacher model using the student's weights that consequently leads to faster and more robust convergence. *RemixIT* is the first method that:

- Performs self-supervised learning using only in-domain mixture datasets and OOD noise sources (e.g. MixIT pre-trained teacher with an OOD dataset).
- Yields state-of-the-art results on several unsupervised and semi-supervised denoising tasks without the need of clean speech waveforms or ad-hoc filtering procedures.
- Has strong theoretical and empirical evidence of why it works under various noise levels.
- Is able to leverage huge amounts of unsupervised data and generalize in diverse training and adaptation scenarios.

II. RemixIT Method

RemixIT trains a speech enhancement model to isolate the clean speech signal from its noisy observation. In general, we train a separation model f which outputs M source waveforms for each input noisy speech recording with T time-domain samples. Thus, given as input a batch of B input waveforms $\mathbf{x} \in \mathbb{R}^{B \times T}$ the network estimates all sound sources:

$$\widehat{\mathbf{s}}, \widehat{\mathbf{n}} = f(\mathbf{x}; \boldsymbol{\theta}), \quad \mathbf{x} = \mathbf{s} + \sum\nolimits_{i=1}^{M-1} [\mathbf{n}]_i = \widehat{\mathbf{s}} + \sum\nolimits_{i=1}^{M-1} [\widehat{\mathbf{n}}]_i, \quad (1)$$

where $\widehat{\mathbf{s}}, \mathbf{s} \in \mathbb{R}^{B \times T}$, $\widehat{\mathbf{n}}, \mathbf{n} \in \mathbb{R}^{(M-1) \times B \times T}$, $\boldsymbol{\theta}$ are: the estimated speech signal, the clean speech target, the estimated noise signal, the noise target and the parameters of the model, respectively. We force the estimated sources $\widehat{\mathbf{s}}$ and $\widehat{\mathbf{n}}$ to add up to the initial input mixtures $\mathbf{x}$ by using a mixture consistency projection layer [45]. We portray the inference and self-training aspects of *RemixIT* in Figure 1, summarize it in Algorithm 1 and analyze it in depth in Section II-C. For completion, we highlight how *RemixIT* differs from fully supervised training (assumes access to clean in-domain speech) and previous state-of-the-art semi-supervised training methods (MixIT assumes access to isolated in-domain noise recordings) in Sections II-A and II-B, respectively.

Algorithm 1: REMIXIT for the noisy dataset $\mathcal{D}_m$.

```
$\boldsymbol{\theta}_{\mathcal{T}}^{(0)} \leftarrow$ PRETRAIN_TEACHER($f_{\mathcal{T}}, \mathcal{D}'$)
$\boldsymbol{\theta}_{\mathcal{S}} \leftarrow$ INITIALIZE_STUDENT($f_{\mathcal{S}}$)
for $k = 0$; $k$++; while $k <= K$ do
    for SAMPLE_BATCH $\mathbf{m} \in \mathcal{D}_m, \;\; \mathbf{m} \in \mathbb{R}^{B \times T}$ do
        $\widetilde{\mathbf{s}}, \widetilde{\mathbf{n}} \leftarrow f_{\mathcal{T}}(\mathbf{m}; \boldsymbol{\theta}_{\mathcal{T}}^{(k)})$ // Teacher's estimates
        $\widetilde{\mathbf{m}} = \widetilde{\mathbf{s}} + \mathbf{P}\widetilde{\mathbf{n}}$ // Bootstrapped remixing
        $\widehat{\mathbf{s}}, \widehat{\mathbf{n}} \leftarrow f_{\mathcal{S}}(\widetilde{\mathbf{m}}; \boldsymbol{\theta}_{\mathcal{S}}^{(k)})$ // Student's estimates
        $\mathcal{L}_{\text{RemixIT}} = \sum_{b=1}^{B} \left[ \mathcal{L}(\widehat{s}_b, \widetilde{s}_b) + \mathcal{L}(\widehat{n}_b, [\mathbf{P}\widetilde{\mathbf{n}}]_b) \right]$
        $\boldsymbol{\theta}_{\mathcal{S}} \leftarrow$ UPDATE_STUDENT($\boldsymbol{\theta}_{\mathcal{S}}, \nabla_{\boldsymbol{\theta}_{\mathcal{S}}} \mathcal{L}_{\text{RemixIT}}$)
    end
    $\boldsymbol{\theta}_{\mathcal{T}}^{(k+1)} \leftarrow$ UPDATE_TEACHER($\boldsymbol{\theta}_{\mathcal{T}}^{(k)}, \boldsymbol{\theta}_{\mathcal{S}}$)
end
```

A. Supervised training

Supervised training is the straightforward way of training speech enhancement models. It assumes access to both in-domain clean speech recordings, $\mathbf{s} \sim \mathcal{D}_s$, as well as noise sources drawn from $\mathbf{n} \sim \mathcal{D}_n$. Synthetic mixtures are generated at each training step $\mathbf{m} = \mathbf{s} + \mathbf{n}$, by sampling a batch of clean speech recordings $\mathbf{s} \sim \mathcal{D}_s$ and a batch of isolated noise samples $\mathbf{n} \sim \mathcal{D}_n$, which are then fed to the separation model f. For a sampled batch of B input mixtures, the model predicts $M = 2$ sources for each input mixture ($\widehat{\mathbf{s}}$, $\widehat{\mathbf{n}} = f(\mathbf{x}; \boldsymbol{\theta})$) and the following targeted loss function is minimized:

$$\mathcal{L}_{\text{Supervised}} = \sum_{b=1}^{B} \left[\mathcal{L}(\widehat{s}_b, s_b) + \mathcal{L}(\widehat{n}_b, n_b) \right], \tag{2}$$

where $\mathcal{L}$ is any desired signal-level loss function used to penalize the reconstruction error between the estimates and their corresponding targets. However, this training process is completely dependent on the availability of clean speech and noise sources to capture the real-world mixture distribution, making the model vulnerable to a performance decline under unseen test conditions. This necessitates the development of SSL and adaptation techniques for speech enhancement.

B. Mixture invariant training (MixIT)

MixIT [28] is a simple yet effective idea for training a separation model using artificial mixtures of mixtures (MoMs). In essence, MixIT assumes availability of two sources of data during training, $\mathcal{D}_m$ which consists of mixtures of speech and a noise source and $\mathcal{D}'_n$ which contains noise recordings from a single noise source. The training process boils down to sampling a batch of noisy speech recordings $\mathbf{m} \sim \mathcal{D}_m$ (where $\mathbf{m} = \mathbf{s} + \mathbf{n}^{(1)}$), and mixing them with another batch of isolated noise recordings $\mathbf{n}^{(2)} \sim \mathcal{D}'_n$. Note that the true noise distribution of the real-world $\mathcal{D}^*_n$ ($\mathbf{n}^{(1)} \sim \mathcal{D}^*_n$) is unknown and not necessarily same as the one available $\mathcal{D}'_n$. The separation model $f_{\mathcal{M}}$ is trained using the synthetic batch of input-MoMs $\mathbf{x} = \mathbf{s} + \mathbf{n}_1 + \mathbf{n}_2$ and tries to reconstruct $M = 3$ sources $\widehat{\mathbf{s}}$, $\widehat{\mathbf{n}}^{(1)}$, $\widehat{\mathbf{n}}^{(2)} = f_{\mathcal{M}}(\mathbf{x}; \boldsymbol{\theta}_{\mathcal{M}})$, by minimizing the following permutation invariant [46] loss function:

$$\mathcal{L}_{\text{MixIT}}^{(b)} = \min_{\boldsymbol{\pi} \in \mathcal{P}} \left[\mathcal{L}(\widehat{s}_b + \widehat{n}_b^{(\pi_1)}, m_b) + \mathcal{L}(\widehat{n}_b^{(\pi_2)}, n_b) \right], \tag{3}$$

where b is the batch's index and $\mathcal{P} := \{(1,2),(2,1)\}$ is the set of permutations between the model's noise output slots. One could also use a probabilistic assignment of the noise estimates $\widehat{n}_b^{(\pi_1)}$, $\widehat{n}_b^{(\pi_2)}$ to avoid emerging problems with the complex permutation invariant landscapes [47].

If the noise sources are independent from each other and the clean speech component, then the model can learn to minimize this loss by reconstructing the mixture using its first estimated slot and either one of the two noise slots available. Although MixIT has been proven effective for various simulated speech enhancement setups [29], [30], the assumption about having access to a diverse set of in-domain noise recordings from $\mathcal{D}'_n$ which aptly captures the true distribution of the present background noises $\mathcal{D}^*_n$ make it impractical for many real-world settings. To this end, other works [24], [31] have tried to deal with the distribution shift between the on-hand noise dataset $\mathcal{D}_n$ and the actual noise distribution $\mathcal{D}^*_n$ in order to avoid the need of in-domain noise samples. Specifically, [31] proposes to use extra noise injection from an OOD distribution and in [24] ASR and disentanglement losses have been proposed. However, the performance of the former method still depends heavily the level of distribution shift between the actual noise distribution $\mathcal{D}^*_n$ and $\mathcal{D}_n$ while the latter method is more restrictive since it requires large pre-trained ASR models.

C. RemixIT: Self-training with bootstrapped remixing

In contrast to the aforementioned two training procedures which require in-domain ground truth signals (e.g. supervised training requires clean speech samples from $\mathbf{s} \sim \mathcal{D}_s$ as well as access to in-domain noise recordings sources drawn from $\mathbf{n} \sim \mathcal{D}_n$ while MixIT requires only isolated in-domain noise waveforms), *RemixIT* does not depend on any other in-domain information besides the mixture dataset $\mathcal{D}_m$. Specifically, our method utilizes a student-teacher framework where the teacher's noise estimates are randomly permuted in a mini-batch sense and remixed with the teacher's speech estimates to create bootstrapped mixtures. A student model is trained using as input the bootstrapped mixtures and regressing over the teacher's pseudo-target signals using a regular supervised loss at every optimization step (for a succinct description of the training procedure please see Algorithm 1). *RemixIT* also enjoys a continual refinement of the noisy pseudo-target signals, after a few optimization steps, where the student model weights are used to update the teacher network as it is illustrated in Figure 1.

1) RemixIT's teacher-student framework: For the initial teacher model, *RemixIT* can use any speech enhancement model pre-trained on an OOD dataset $\mathcal{D}'$ which outputs the speech component and one or more noise estimated waveforms (see specification in Equation 1). To this end, *RemixIT* materializes into semi-supervised domain adaptation if the teacher was trained using a supervised loss and into a SSL training scheme if the teacher was trained using MixIT.

Formally, given an batch of in-domain noisy mixtures $\mathbf{m} = \mathbf{s} + \mathbf{n} \in \mathbb{R}^{B \times T}, \;\; \mathbf{m} \sim \mathcal{D}_m$, the teacher model estimates the speech and the noise components as follows:

$$\widetilde{\mathbf{s}}, \, \widetilde{\mathbf{n}} = f_{\mathcal{T}}(\mathbf{x}; \boldsymbol{\theta}_{\mathcal{T}}^{(k)}), \;\; \mathbf{m} = \widetilde{\mathbf{s}} + \sum_{i=1}^{M-1} [\mathbf{n}']_i \tag{4}$$

where $\boldsymbol{\theta}_{\mathcal{T}}^{(k)}$ denotes the parameters of the teacher model at the k-th optimization step. The second equation holds because we enforce mixture consistency. A MixIT pre-trained model would estimate $M = 3$ sources and we can easily get a consolidated noise estimate by summing the two latter noise estimated waveforms, namely, $\widetilde{\mathbf{n}} = \sum_{i=1}^{M-1}[\mathbf{n}']_i$. Notice that the teacher model $f_{\mathcal{T}}$ does not need to be identical through the whole training process and could be updated using any user-specified protocol which results in a separation model that respects the constraints defined in Equation 4. The teacher's estimates within a batch of size B are used to generate the bootstrapped mixtures $\widetilde{\mathbf{m}}$ by remixing the estimated speech and noise sources in a random order:

$$\widetilde{\mathbf{m}} = \widetilde{\mathbf{s}} + \widetilde{\mathbf{n}}^{(\mathbf{P})} \in \mathbb{R}^{B \times T}, \;\; \widetilde{\mathbf{n}}^{(\mathbf{P})} = \mathbf{P}\widetilde{\mathbf{n}}, \;\; \mathbf{P} \sim \mathbf{\Pi}_{B \times B}, \tag{5}$$

where $\mathbf{P}$ is drawn uniformly from the set of all $B \times B$ permutation matrices and is used to produce the permuted noise sources $\widetilde{\mathbf{n}}^{(\mathbf{P})}$. The original teacher's speech estimates $\widetilde{\mathbf{s}}$ and the permuted noise sources $\widetilde{\mathbf{n}}^{(\mathbf{P})}$ are now used as target pairs to train the student model $f_{\mathcal{S}}$ on the newly generated batch of bootstrapped mixtures $\widetilde{\mathbf{m}}$ as shown below:

$$\begin{gathered}\widehat{\mathbf{s}},\, \widehat{\mathbf{n}} = f_{\mathcal{S}}(\widetilde{\mathbf{m}}; \boldsymbol{\theta}_{\mathcal{S}}^{(k)}), \;\; \widehat{\mathbf{s}},\, \widehat{\mathbf{n}} \in \mathbb{R}^{B \times T} \\ \mathcal{L}_{\text{RemixIT}}^{(b)} = \mathcal{L}(\widehat{\mathbf{s}}_b, \widetilde{\mathbf{s}}_b) + \mathcal{L}(\widehat{\mathbf{n}}_b, \mathbf{P}\widetilde{\mathbf{n}}_b), \;\; b \in \{1, \dots, B\}. \\ \mathcal{L}_{\text{RemixIT}} = \sum_{b=1}^{B} \left[\mathcal{L}(\widehat{\mathbf{s}}_b, \widetilde{\mathbf{s}}_b) + \mathcal{L}(\widehat{\mathbf{n}}_b, \mathbf{P}\widetilde{\mathbf{n}}_b)\right]\end{gathered} \tag{6}$$

The loss function used is similar to a supervised setup (see Equation 2) but instead of ground-truth clean source waveforms, we use the noisy estimates, $\widetilde{\mathbf{s}}$ and $\widetilde{\mathbf{n}}^{(\mathbf{P})}$, provided by the teacher network. If the signal-level loss function $\mathcal{L}$ also minimizes the Euclidean norm between the estimated signals and the target signals, the proposed cost function $\mathcal{L}_{\text{RemixIT}}$ enjoys several convergence properties which enable our method to learn in a robust SSL fashion even in cases where the teacher's estimates are not close to the ground-truth source waveforms (see Section II-C2).

Lastly, *RemixIT* refines the estimates of the teacher network $f_{\mathcal{T}}$ using the weights from the latest available student models. The continual update protocols used in this study are the sequential and the running moving average update protocols which are explained in detail in Section III-C.

2) Error analysis under the Euclidean norm: In each optimization step, *RemixIT* tries to minimize a signal-level loss function between the student's estimates and the teacher's pseudo-targets. Since we are mostly interested in denoising, we focus on the speech estimates of the teacher and the student networks with initial mixtures $\mathbf{M}$ and the bootstrapped mixtures $\widetilde{\mathbf{M}}$ as inputs, respectively. These estimates can also be expressed in the following way as random variables:

$$\begin{gathered}\widetilde{\mathbf{S}} = f_{\mathcal{T}}^{(\widetilde{s})}(\mathbf{M} = \mathbf{S} + \mathbf{N};\, \boldsymbol{\theta}_{\mathcal{T}}^{(k)}), \;\; \mathbf{M} \sim \mathcal{D}_m \\ \widehat{\mathbf{S}} = f_{\mathcal{S}}^{(\widehat{s})}\,(\widetilde{\mathbf{M}} = \widetilde{\mathbf{S}} + \widetilde{\mathbf{N}}^{(\mathbf{P})}; \boldsymbol{\theta}_{\mathcal{S}}^{(k)}).\end{gathered} \tag{7}$$

Now, the teacher's $\widetilde{\mathbf{R}}_{\mathcal{T}}$ and student's $\widehat{\mathbf{R}}_{\mathcal{S}}$ errors w.r.t. the initial clean targets $\mathbf{S}$ are the following conditional probabilities:

$$\begin{gathered}\widetilde{\mathbf{R}}_{\mathcal{T}} = \widetilde{\mathbf{S}} - \mathbf{S}, \quad \widetilde{\mathbf{R}}_{\mathcal{S}} = \widehat{\mathbf{S}} - \mathbf{S} \\ \widetilde{\mathbf{R}}_{\mathcal{T}} \sim P(\widetilde{\mathbf{R}}_{\mathcal{T}} | \mathbf{S}, \mathbf{N}), \;\; \widehat{\mathbf{R}}_{\mathcal{S}} \sim P(\widehat{\mathbf{R}}_{\mathcal{S}} | \widetilde{\mathbf{S}}, \widetilde{\mathbf{N}}, \mathbf{P}).\end{gathered} \tag{8}$$

Using a signal-level loss $\mathcal{L}$ that minimizes the squared error between the estimated and the target signals in Equation 6 and assuming unit-norm estimated and target signals $||s|| = ||\widetilde{s}|| = ||\widehat{s}|| = 1$, *RemixIT* loss function becomes equivalent to minimizing the following expression:

$$\begin{aligned}\mathcal{L}_{\text{RemixIT}} \propto \mathbb{E}[||\widehat{\mathbf{S}} - \widetilde{\mathbf{S}}||_2^2] = \mathbb{E}[||(\widehat{\mathbf{S}} - \mathbf{S}) - (\widetilde{\mathbf{S}} - \mathbf{S})||_2^2] \\ = \underbrace{\mathbb{E}\left[||\widehat{\mathbf{R}}_{\mathcal{S}}||_2^2\right]}_{\text{Supervised Loss}} + \underbrace{\mathbb{E}\left[||\widetilde{\mathbf{R}}_{\mathcal{T}}||_2^2\right]}_{\text{Constant w.r.t. } \boldsymbol{\theta}_{\mathcal{S}}} -2\underbrace{\mathbb{E}\left[\langle \widehat{\mathbf{R}}_{\mathcal{S}}, \widetilde{\mathbf{R}}_{\mathcal{T}} \rangle\right]}_{\text{Errors' correlation}}\end{aligned} \tag{9}$$

Ideally, this loss could lead to the same optimization objective with a supervised setup if the last inner-product term was zero since the middle term becomes zero when computing the gradient w.r.t. the student's parameters $\boldsymbol{\theta}_{\mathcal{S}}$. $\langle \widehat{\mathbf{R}}_{\mathcal{S}}, \widetilde{\mathbf{R}}_{\mathcal{T}} \rangle = 0$ could be achieved if the teacher produced outputs indistinguishable from the clean target signals or the conditional error distributions in Equation 8 were independent. Intuitively, as we continually update the teacher model and refine its estimates, we minimize the norm of the teacher error which leads to higher fidelity reconstruction from the student (for further analysis of how the student learns to perform better than its teacher and for experimental validation of this claim we refer the reader to Section IV-D).

Additionally, the bootstrapped remixing process forces the errors to be more uncorrelated since the student tries to reconstruct the same clean speech signals $\mathbf{s}$, similar to its teacher, but under a different mixture distribution. Formally, the student tries to reconstruct $\mathbf{s}$ when observing the bootstrapped mixtures $\widetilde{\mathbf{m}} = \widetilde{\mathbf{s}} + \widetilde{\mathbf{n}}^{(\mathbf{P})}$ while the teacher tries to reconstruct $\mathbf{s}$ only from the initial input mixtures $\mathbf{m} = \mathbf{s} + \mathbf{n}$. This phenomenon becomes apparent if we focus on the reconstruction of a single speech signal s^* from the teacher and the student networks. In essence, we use the teacher network to provide an estimated $\widetilde{s}^*$ from the corresponding mixture m^* and some perturbed noise sources $\widetilde{n}_b'$, $\forall b \in \{1, \dots, B\}$ to create bootstrapped mixtures:

$$\begin{aligned}\widetilde{s}^*, \widetilde{n}^* &= f_{\mathcal{T}}(m^* = s^* + n^*; \boldsymbol{\theta}_{\mathcal{T}}) \\ \widetilde{s}_b', \widetilde{n}_b' &= f_{\mathcal{T}}(m_b' = s_b' + n_b'; \boldsymbol{\theta}_{\mathcal{T}}) \\ \widetilde{m}_b' &= \widetilde{s}^* + \widetilde{n}_b', \;\; \forall b \in \{1, \dots, B\}.\end{aligned} \tag{10}$$

In the student-training phase, we perform inference using the student network $f_{\mathcal{S}}$ on the batch of the aforementioned bootstrapped mixtures $\widetilde{m}_b'$. Because *RemixIT*'s loss is computed under expectation (Equation 9), we can rearrange the order of batches that the student network sees. Thus, we focus on the learning aspect of the student network for the batch of bootstrapped mixtures above (Equation 10) and rewrite the last error correlation term as follows:

$$\begin{aligned}\mathbb{E}\left[\langle \widehat{\mathbf{R}}_{\mathcal{S}}, \widetilde{\mathbf{R}}_{\mathcal{T}} \rangle\right] \approx \mathbb{E}\left[\frac{1}{B}\sum_{b=1}^{B}\left(\widehat{s}_b - s^*\right)^{\mathrm{T}}\left(\widetilde{s}^* - s^*\right)\right] \\ = \mathbb{E}[(\widetilde{s}^* - s^*)^{\mathrm{T}}\underbrace{\frac{1}{B}\sum_{b=1}^{B}\left(f_{\mathcal{S}}^{(\widehat{s})}(\widetilde{s}^* + \widetilde{n}_b'; \boldsymbol{\theta}_{\mathcal{S}}) - s^*\right)}_{\text{Empirical mean student error}}]\end{aligned} \tag{11}$$

The premise is that if the student sees a wide variety of bootstrapped mixtures which have been generated using the same teacher's speech estimate $\widetilde{s}^*$, then the mean interference error produced by injecting noisy teacher's estimates $\widetilde{n}_b'$ would go to zero under expectation. We prove this claim under ideal conditional independence of the student error vectors and infinite bootstrapped mixtures in Theorem II.1. In practice, the student could still minimize the errors' correlation term and still be able to learn from mixtures when the teacher performs poorly (please see Section IV-E which gives an empirical analysis of our claim).

Theorem II.1. *Assuming a differentiability of the loss functions, access to infinite bootstrapped mixtures $B \rightarrow \infty$ generated by the teacher network f_{T}, and conditional independence of the student errors given the same teacher speech pseudo-target ($f_{\mathcal{S}}^{(\widehat{s})}(\widetilde{s}^* + \widetilde{n}_i'; \boldsymbol{\theta}_{\mathcal{S}}) - \widetilde{s}^* \perp f_{\mathcal{S}}^{(\widehat{s})}(\widetilde{s}^* + \widetilde{n}_j'; \boldsymbol{\theta}_{\mathcal{S}}) - \widetilde{s}^*$ with $i \neq j$), then the gradients of RemixIT's loss function w.r.t. the student network weights $\boldsymbol{\theta}_{\mathcal{S}}$) converge to the ones provided by an oracle supervised loss $\nabla_{\boldsymbol{\theta}_{\mathcal{S}}} \mathcal{L}_{\mathrm{RemixIT}} \approx \nabla_{\boldsymbol{\theta}_{\mathcal{S}}} \mathcal{L}_{\mathrm{Supervised}}$*

Proof. Combining the definitions of the loss functions from Equations 2 and 6, their difference can be expressed as:

$$\mathcal{L}_{\mathrm{RemixIT}} - \mathcal{L}_{\mathrm{Supervised}} = \mathbb{E}\left[||\widetilde{\mathbf{R}}_{\mathcal{T}}||_2^2 - 2 \langle \widehat{\mathbf{R}}_{\mathcal{S}}, \widetilde{\mathbf{R}}_{\mathcal{T}} \rangle \right]. \quad (12)$$

Following the same analysis with Section II-C2, for each target speaker waveform s^*, we use the estimates of the teacher model for the target speech waveform $\widetilde{s}^*$ in an input mixture m^* and for randomly sampled noise sources $\widetilde{n}_b'$ in the corresponding mixtures m_b' as in Equations 5 to produce bootstrapped mixtures $\widetilde{m}_b' = \widetilde{s}^* + \widetilde{n}_b', \;\; \forall b$. Thus, the student estimates some speech waveform $\widehat{s}_b$ for each input bootstrapped mixture $\widetilde{m}_b'$ and the latter term of the error correlation can be written as follows:

$$\begin{aligned} & \mathbb{E}\left[\langle \widehat{\mathbf{R}}_{\mathcal{S}}, \widetilde{\mathbf{R}}_{\mathcal{T}} \rangle \right] = \mathbb{E}\left[(\widetilde{s}^* - s^*)^{\mathrm{T}} \frac{1}{B} \textstyle\sum_{b=1}^{B} (\widehat{s}_b - s^*) \right] = \\ & \mathbb{E}\left[(\widetilde{s}^* - s^*)^{\mathrm{T}} \frac{1}{B} \textstyle\sum_{b=1}^{B} \left[(\widehat{s}_b - \widetilde{s}^*) + (\widetilde{s}^* - s^*) \right] \right] = \\ & \mathbb{E}\left[||\widetilde{\mathbf{R}}_{\mathcal{T}}||_2^2 \right] + \mathbb{E}\left[(\widetilde{s}^* - s^*)^{\mathrm{T}} \frac{1}{B} \textstyle\sum_{b=1}^{B} (\widehat{s}_b - \widetilde{s}^*) \right]. \end{aligned} \quad (13)$$

However, the error between each pseudo-target provided by the teacher student $\widetilde{s}^* = f_{\mathcal{T}}^{(\widetilde{s})}(m^* = s^* + n^*; \boldsymbol{\theta}_{\mathcal{T}})$ and the estimated speech signal by the student $\widehat{s}_b' = f_{\mathcal{S}}^{(\widehat{s})}(\widetilde{s}^* + \widetilde{n}_b'; \boldsymbol{\theta}_{\mathcal{S}})$ is bounded for any masked-based network operating on some linear bases (we use a linear encoder/decoder as specified in Section III-B) [48]. Formally, assuming that an the encoded representation of the input bootstrapped mixture m' is $v' = \mathbb{P} \cdot m'$, then the latent representation of a signal estimate is $\hat{v} = \widehat{\mathbf{M}} \odot (\mathbb{P} \cdot m')$. Thus, the l_2 error is bounded by:

$$\begin{aligned} & ||\widehat{s}_b' - \widetilde{s}^*|| = ||(\widehat{\mathbf{M}}_b' - \widetilde{\mathbf{M}}^*) \odot (\mathbb{P} \cdot \widetilde{m}_b')|| \\ & \leq \max_{\widetilde{s}^*, \widehat{s}_b', \widetilde{n}_b'} \left[\sigma_{\max}\{(\widehat{\mathbf{M}}_b' - \widetilde{\mathbf{M}}^*) \odot \mathbb{P}\} \cdot ||\widetilde{s}^* + \widetilde{n}_b'|| \right] = \widehat{C}, \end{aligned} \quad (14)$$

where $\sigma_{\max}\{(\widehat{\mathbf{M}}_b' - \widetilde{\mathbf{M}}^*) \odot \mathbb{P}\} < \infty$ denotes the maximum singular value of the masked unrolled synthesis-basis matrix $\mathbb{P}$ and $||\widetilde{s}^* + \widetilde{n}_b'|| < \infty$ is the energy of the bounded-norm bootstrapped mixtures. Similarly, the teacher error is also bounded by some real value $||\widetilde{s}^* - s^*|| \leq \widetilde{C} < \infty, \;\; \forall s^*$.

Thus, by combining the above inequalities with Equations 12 and 13, we conclude that at the limit, the difference of the loss functions converges to a value which is constant with respect to the student network's parameters $\boldsymbol{\theta}_{\mathcal{S}}$ as shown next:

$$\begin{aligned} & \lim_{B \rightarrow \infty} \left[\mathcal{L}_{\mathrm{RemixIT}} - \mathcal{L}_{\mathrm{Supervised}} \right] = -\mathbb{E}\left[||\widetilde{\mathbf{R}}_{\mathcal{T}}||_2^2 \right] \\ & + \lim_{B \rightarrow \infty} \mathbb{E}\left[(\widetilde{s}^* - s^*)^{\mathrm{T}} \frac{1}{B} \sum_{b=1}^{B} (\widehat{s}_b - \widetilde{s}^*) \right] = \\ & -\mathbb{E}\left[||\widetilde{\mathbf{R}}_{\mathcal{T}}||_2^2 \right] + \lim_{B \rightarrow \infty} \mathop{\mathbb{E}}_{\widetilde{s}^*}\left[(\widetilde{s}^* - s^*)^{\mathrm{T}} \mu(\widetilde{s}^*) \right]. \end{aligned} \quad (15)$$

where the last step comes from the application of the central limit theorem since by assumption the student estimates' errors are i.i.d. and bounded, thus, the sample mean converges in distribution to a normal distribution with mean equal to the mean student error $\mu(\widetilde{s}^*)$ given the corresponding teacher's speech estimate $\widetilde{s}^*$. All parts of the right hand-side of the above equation are constant w.r.t. the student network parameters $\boldsymbol{\theta}_{\mathcal{S}}$ which we try to optimize and thus by applying the gradient operator we can conclude that $\nabla_{\boldsymbol{\theta}_{\mathcal{S}}} \mathcal{L}_{\mathrm{RemixIT}} \approx \nabla_{\boldsymbol{\theta}_{\mathcal{S}}} \mathcal{L}_{\mathrm{Supervised}}$. One could make this theorem even more applicable to real-world settings where the student errors given different bootstrapped mixtures from the initial teacher estimate $\widetilde{s}^*$ are weakly dependent [49] but we defer this derivation to future work. □

III. Experimental Framework

A. Datasets

DNS-Challenge (DNS): The DNSChallenge 2020 benchmark dataset [50] consists of a large collection of clean speech recordings which are mixed with a wide variety of noisy speech samples with 64,649 and 150 pairs of clean speech and noise recordings for training and testing, respectively. DNS is used for showing the effectiveness of the proposed self-training scheme where large amounts of unsupervised training data is available and one needs to improve the performance of a model trained only on limited OOD supervised data.

LibriFSD50K (LFSD): This data collection includes 45,602 and 3,081 mixtures for training and testing, correspondingly. The clean speech samples are drawn from the LibriSpeech [51] corpus and the noise recordings are taken from FSD50K [52] representing a set of almost 200 classes of background noises after excluding all the human-made sounds from the AudioSet ontology [53]. A detailed recipe of the dataset generation process is presented in [30]. LFSD becomes an ideal candidate for semi-supervised/SSL teacher pre-training on OOD data given its mixture diversity.

WHAM!: The generation process for this dataset produces 20,000 training noisy-speech pairs and 3,000 test mixtures from the initial WHAM! [54] dataset and has been identical to the procedure followed in [30] with active noise sources mixed at an average of −1.3dB input SNR. The set of background noises in WHAM! is limited to 10 classes of urban sounds.

VCTK: The VCTK dataset proposed in [55] includes 586 synthetically generated noisy test mixtures, where a speech sample from the VCTK speech corpus [56] is mixed with an isolated noise recording from the DEMAND [57]. The VCTK and DNS test partitions are used to illustrate the effectiveness of *RemixIT* under a restrictive scenario zero-shot domain adaptation with limited data to perform self-training.

B. Speech enhancement model

In the supervised and *RemixIT* training recipes, the student has $M = 2$ output slots and always estimates the speech component and the noise source. For the models which are trained with MixIT, we increase the number of output slots to $M = 3$ to estimate the additional noise component. *RemixIT* is independent of the choice of the speech-enhancement model architecture as long as the latter estimates both speech and noise components of the input mixture.

Our model's choice was based on obtaining adequate quality of speech reconstruction with low computational and memory requirements (see Table I for a head-to-head comparison in a supervised in-domain training setup with the previous state-of-the-art model). To this end, we used the Sudo rm -rf [58] architecture with the more sparse computation blocks using shared sub-band processing via group communication [59]. The selected network has shown to provide high-quality source estimates under speech enhancement [30] as well as sound separation [60] tasks while significantly reducing the model's size. We consider the selected architecture with a default encoder/decoder with 512 basis 41 filter taps and a hop-size of 20 time-samples, a depth of $U = 8$ U-ConvBlocks and the same parameter configurations as used in [30]. In the sequential update protocol we increase the depth of the new student networks every 20 epochs from 8 to 16 and finally 32.

C. RemixIT's teacher update protocols configurations

RemixIT refines the estimates of the student network based on unsupervised and semi-supervised teachers pre-trained on an OOD dataset but also has the capability of repeatedly updating the teacher network to learn from higher-quality source estimates. In our experiments, we evaluate the proposed method under various online teacher updating protocols after k training epochs. Specifically, we consider the following:
Static teacher: The teacher is frozen throughout the training process, for all optimization steps $\boldsymbol{\theta}_{\mathcal{T}}^{(k)} = \boldsymbol{\theta}_{\mathcal{T}}^{(0)}, \ \forall k$.
Sequentially updated teacher: Every 20 epochs or equivalently $K = {}^{20 \times |\mathcal{D}_m|}/_{B}$ optimization steps, where ${}^{|\mathcal{D}_m|}/_{B}$ is the number of batches per-training epoch, we replace the teacher with the latest student, namely, $\mathring{\boldsymbol{\theta}}_{\mathcal{T}}^{(k \bmod K)} := \boldsymbol{\theta}_{\mathcal{S}}^{(k \bmod K)}$.
Exponentially moving average teacher: The teacher is gradually updated after every epoch using an exponential moving average scheme $\bar{\boldsymbol{\theta}}_{\mathcal{T}}^{(j+1)} := \gamma \boldsymbol{\theta}_{\mathcal{S}}^{(j)} + (1 - \gamma) \bar{\boldsymbol{\theta}}_{\mathcal{T}}^{(j)}, \ \forall k$ with $\gamma = 0.01$, where j is a multiple of ${}^{|\mathcal{D}_m|}/_{B}$.

D. Training and evaluation details

For the semi-supervised and unsupervised *RemixIT*'s teachers we pre-train the corresponding models following the supervised training process (Section II-A) and MixIT (Section II-B), respectively. Although *RemixIT* can theoretically work with any valid signal-level loss functions (Equations 3, 6), we choose the negative scale-invariant signal to distortion ratio (SI-SDR) [61] for training all models:

$$\mathcal{L}(\widehat{y}, y) = -\text{SI-SDR}(\widehat{y}, y) = -20 \log_{10}\left({}^{\|\alpha y\|}/_{\|\alpha y - \widehat{y}\|} \right). \quad (16)$$

$\alpha = \widehat{y}^\top y / \|y\|^2$ makes the loss invariant to the scale of the estimated source $\widehat{y}$ and the target signal y. By setting $\alpha = 1$, SI-SDR becomes equivalent with SNR. We train all models using the Adam optimizer [62] with a batch size of $B = 2$ and an initial learning rate of 10^{-3} which is divided by 2 every 6 epochs. We fix those hyper-parameters after some early experimentation with the validation set of LFSD. For all experiments, during training we assume that we do not have access to the input SNR distribution and thus, we mix a clean and a noise source without altering their corresponding power ratio. However, for the in-domain supervised training setup with DNS we randomly mix clean speech and noise recordings with SNR from a uniform distribution of $[-2, 20] dB$, which has been shown to be effective for multiple sound separation setups [48], [63], [64]. Finally, we normalize all input mixture waveforms by subtracting their mean and dividing by their standard deviation before feeding them to each model. We train and test models which operate at a 16kHz sampling rate.

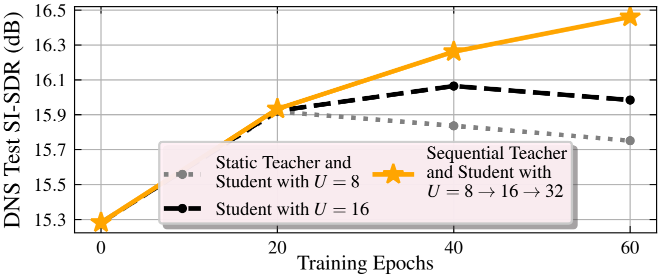


Fig. 2: SI-SDR (dB) performance on DNS test as training progresses using different teacher update protocols. The initial teacher network is shared across the various protocols (a Sudo rm -rf model with $U = 8$ Conv-blocks) and was pre-trained in a supervised way on the WHAM! dataset. The orange solid line denotes the performance of the student model with increasing depth every 20 training epochs $U : 8 \to 16 \to 32$ where we initialize a new student model and replace the teacher model with the latest available student. The sequential protocol shows significant gains over the static teacher protocols where the student network has a static architecture throughout training and the initial teacher is not updated ($U = 8$ with gray and $U = 16$ with black dashed lines).

The robustness of all speech enhancement models is measured using the SI-SDR [61], the short-time objective intelligibility (STOI) [65] and the perceptual evaluation of speech quality (PESQ) [66]. We evaluate the model checkpoints after 100 epochs for the pre-trained teachers and the supervised models and after 60 epochs for all the other configurations.

IV. Results and Discussion

A. The need for continual refinement of teacher's estimates

In Figure 2, we show the speech enhancement performance of the student models produced by a static or a sequentially updated teacher every 20 epochs. Unsurprisingly, all protocols behave similarly until the 20th epoch since they use the same initial teacher. In contrast to the frozen teacher protocols, after the 20th epoch, the old teacher is replaced with the newly trained student with U=8 and a new student, with twice as much depth (8 → 16) is initialized. Surprisingly, the sequentially updated teacher protocol keeps teaching a better

Training method and model details		#Model Params (10^6)	Available Training Data (%)						Mean evaluation metrics		
			Clean Speech $\mathcal{D}_s$		Noise $\mathcal{D}_n$		Mixture $\mathcal{D}_m$		SISDR (dB)	PESQ	STOI
			DNS	LFSD	DNS	LFSD	DNS	LFSD			
Input Noisy Mixture		-							9.2	1.58	0.915
MixIT with Sudo rm-rf ($U = 8$)	In-domain noise	0.79			20%		80%		14.4	2.13	0.933
	OOD noise	0.79				20%	100%		14.3	2.02	0.933
	Extra OOD noise [31]	0.79				50%	100%		14.5	2.03	0.930
Unsupervised RemixIT (ours)	Teacher ($U = 8$)	0.79				20%		80%	14.8	2.15	0.940
	Student ($U = 8$)	0.56					100%		15.5	2.27	0.947
	Student ($U : 8 \rightarrow 16 \rightarrow \mathbf{32}$)	0.73					100%		16.0	2.34	0.952
Semi-supervised RemixIT (ours)	Teacher ($U = 8$)	0.56		100%		100%			17.6	2.61	0.958
	Student ($U = 8$)	0.56					100%		17.6	2.52	0.956
	Student ($U : 8 \rightarrow 16 \rightarrow \mathbf{32}$)	0.73					100%		18.0	2.60	0.959
Supervised in-domain training	FullSubNet* [8]	5.6	100%		100%				17.3	2.78	0.961
	Sudo rm -rf [60] ($U = 8$)	0.56	100%		100%				18.6	2.69	0.962
	Sudo rm -rf [60] ($U = 32$)	0.73	100%		100%				**19.7**	**2.95**	**0.971**

TABLE I: Evaluation results for the speech enhancement task on the DNS test set using the proposed *RemixIT* method, MixIT approaches [28], [31] and supervised in-domain training with the Sudo rm -rf model [60] as well as the previous state-of-the-art *FullSubNet* [8] supervised model (*as it was presented in the paper). All teacher and student networks follow the same Sudo rm -rf model [60] architecture with the specified number of U-ConvBlocks ($U = 8$ or $U = 32$). $U : 8 \rightarrow 16 \rightarrow \mathbf{32}$ denotes that we double the depth of the student network every 20 epochs and sequentially update the teacher with the latest available student, the reported number refers to the performance of the student with $U = 32$.

student separation model, even after the 40th epoch, compared to both models produced by the static teachers which saturate for the same number of training steps. Comparing between the students with U=8 and U=16 produced by static teacher models, it is evident that the more expressive student performs better but not on par with the same depth student produced by the sequentially updated teacher protocol. Specifically, both orange-solid and black-dashed lines at the 40-th epoch represent the performance of a student model with the same depth ($U = 16$) but the sequential update protocol clearly outperforms the frozen-teacher protocol. Thus, the combination of the bootstrapped remixing and the continual refinement of the teacher's estimates is key for the significant improvement that *RemixIT* yields. As a result, we have chosen the sequentially updated teacher protocol as the default strategy for *RemixIT*, except for the zero-shot adaptation where we use the exponential average teacher updating scheme because the number of available training mixtures could make the student prone to overfitting if trained from scratch.

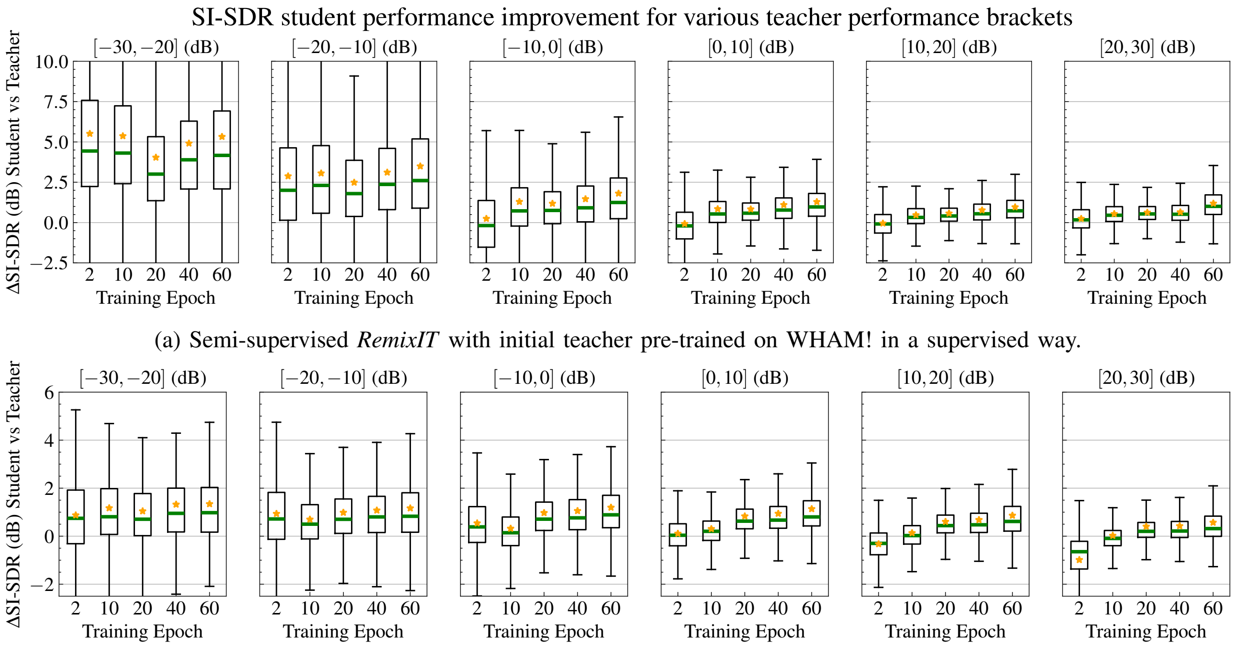
B. Self-supervised and semi-supervised speech enhancement

Table I summarizes the mean speech enhancement performance of *RemixIT* against in-domain and cross-domain supervised and SSL baselines with the same architecture on the DNS test set. Notice that in both semi-supervised and unsupervised cases, the learned *RemixIT*'s student does not assume access to in-domain clean speech nor to noise samples like the previous state-of-the-art SSL speech enhancement algorithms. For instance, SSL *RemixIT*'s teacher pre-training is performed with OOD MixIT by using 80% of the LFSD noisy recordings $\mathcal{D}'_m$ and rest 20% to simulate the isolated noise recordings $\mathcal{D}'_n$, whereas the student is trained solely on the vast amount of training mixtures in the DNS dataset.

Despite the fact that *RemixIT* makes no assumptions about the in-domain distribution of mixtures nor it assumes access to in-domain ground truth source waveforms, it significantly outperforms all the previous state-of-the-art MixIT-like approaches. The unsupervised student learned using the proposed method yields an improvement over the second-best unsupervised method of more than (14.5dB $\rightarrow$ 16.0dB in terms of SI-SDR and 0.02 in terms of STOI) compared against in-domain MixIT and the recently proposed extra noise augmentation where an extra noise source is injected [31]. In the semi-supervised domain adaptation setup, we show that *RemixIT*'s student still provides noticeable improvement over its initial teacher pre-trained in a supervised way assuming access to a smaller but diverse dataset like LFSD. Although we have used the same separation model architecture across our experiments, our method is independent of the model's choice and could be used with models that produce higher quality estimates. However, the bottom three rows in Table I show that the model used in this study achieves state-of-the-art speech enhancement results when trained with in-domain ground-truth sources.

C. Zero-shot domain adaptation

In low-resource training scenarios, the training mixtures in-hand might not be sufficient to train a model from scratch, thus, we show how *RemixIT* can be used as a zero-shot unsupervised domain adaptation algorithm. We perform teacher pre-training on larger OOD datasets and fine-tune a student model using limited in-domain mixtures. At the start of the adaptation process, the student is initialized using the pre-trained teacher's weights $\boldsymbol{\theta}_S^{(0)} := \boldsymbol{\theta}_T^{(0)}$ and we perform *RemixIT* while periodically updating the teacher using the moving average protocol (see Section III-C). The cross-dataset adaptation results are illustrated in Figure 4. The proposed method delivers consistent improvements across datasets and pre-training techniques, up to 0.8dB in terms of SI-SDR over the non-calibrated models. Unsurprisingly, one can notice that the level of improvement is directly impacted by the amount of available noisy mixtures. We postulate that this is the main reason that our method obtains larger (smaller) gains for the adaptation on WHAM! (DNS) test partition which has $3{,}000$ (only 150) mixtures, respectively. However, *RemixIT* performs adequately even in cases where there is a large distribution



(a) Semi-supervised *RemixIT* with initial teacher pre-trained on WHAM! in a supervised way.

(b) Unsupervised *RemixIT* with initial teacher pre-trained on LFSD using MixIT.

Fig. 3: SI-SDR (dB) performance improvement on the training portion of the DNS dataset that a *RemixIT*'s student with a sequentially updated teacher every 20 epochs yields as the training progresses over the initial teacher's estimates. We show that similar learning patterns emerge for different initial teachers pre-trained in a semi-supervised way (top) and an unsupervised way (bottom). The median and the mean ΔSI-SDR are denoted with a solid green line and an orange star, respectively.

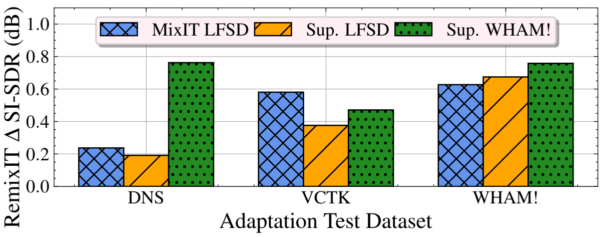


Fig. 4: SI-SDR performance improvement that *RemixIT*'s student yields over its initial OOD pre-trained teacher model for various low-resource adaptation datasets (e.g. DNS, LFSD and WHAM!, from left to right). Both teacher and student models have the exact same Sudo rm -rf architecture ($U = 8$ ConvBlocks) and we use the running mean teacher update protocol. *RemixIT* shows significant improvements against all teacher models used in this study, namely, MixIT pre-training on LFSD (blue/leftmost) and supervised training on LFSD (yellow/middle) and as well as on WHAM! (green/rightmost).

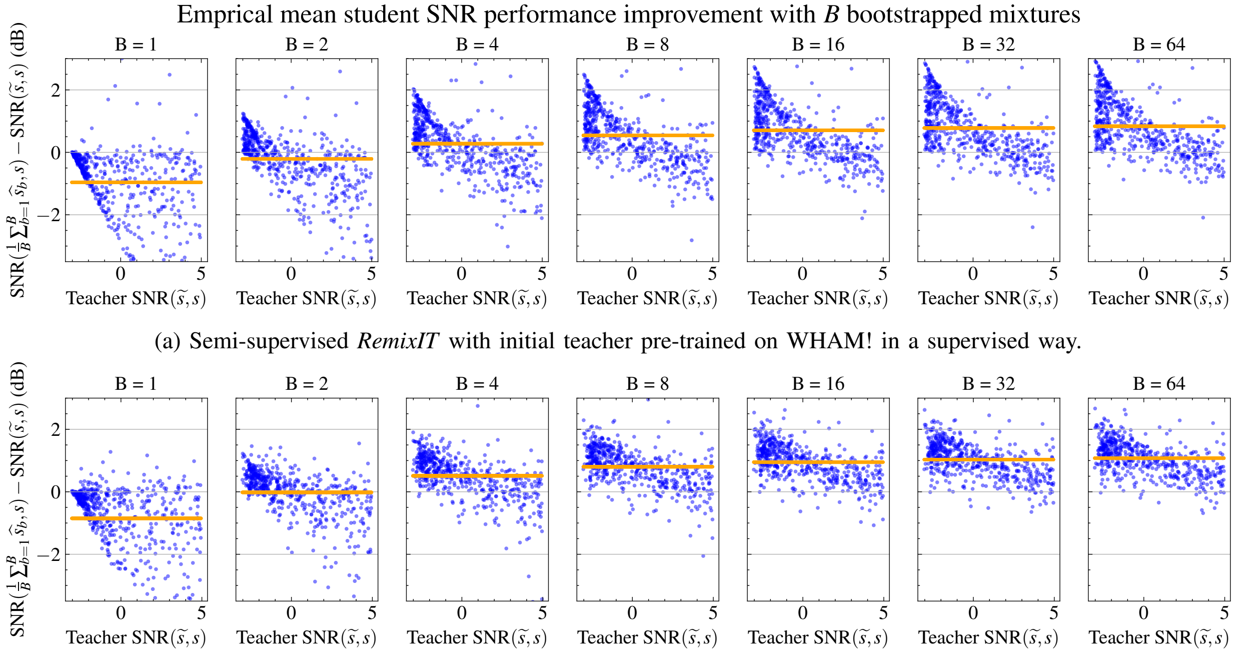
shift between the training and the adaptation-test sets (e.g. WHAM! contains only 10 classes of urban background noises while the DNS dataset is very diverse). Specifically, the significant improvement after adapting a supervised pre-trained model on WHAM! to the 150 mixtures of the very diverse DNS set, indicates the effectiveness of *RemixIT* under really challenging zero-shot learning conditions.

D. Student learning progression

We analyze how a student speech enhancement model trained with *RemixIT* on the DNS train set refines its estimates as the training progresses and how it compares against its initial teacher. In Figure 3, we showcase the improvement obtained in terms of SI-SDR for various teachers and their performance brackets under a sequentially updated teacher every 20 epochs using the parameters from the student network. Note that the student is gradually learning to perform better than the initial teacher network in the regions where the latter performs better (rightmost plots row-wise) even if producing improvement over really good estimates (e.g. higher than 15dB) becomes harder. Thus, it becomes evident that the continual self-training scheme of *RemixIT* where the teacher network is updated using the latest student's weights is key to a larger performance boost. The result holds for both OOD supervised and MixIT teachers and is on par-with our theoretical analysis in Section II-C2 where we show how a better teacher helps the error correlation term of *RemixIT*'s loss function to diminish and resemble supervised training. In contrast, for the low performing brackets ($[-30, -10]$dB in terms of teacher SI-SDR (dB)), the student does not learn how to further increase its performance, even if it regresses over the estimated waveforms of updated teachers. The emergence of this learning pattern necessitates the discovery of more robust self-training algorithms which can recover from cases where the teacher network provides extremely noisy estimates.

E. Robust learning with very noisy teacher's estimates

We investigate the robustness of *RemixIT* in cases where the teacher model outputs a low quality speech estimate $\widetilde{s}$. Building upon the analysis performed in Section II-C2, we reiterate on how important is for the student to be trained



(a) Semi-supervised *RemixIT* with initial teacher pre-trained on WHAM! in a supervised way.

(b) Unsupervised *RemixIT* with initial teacher pre-trained on LFSD using MixIT.

Fig. 5: Distribution of SNR improvement (dB) on the DNS training set that the empirical mean *RemixIT*'s student after 10 training epochs (Equation 11) yields over its initial teacher in regions where the latter performs poorly. The solid orange line denotes the mean SNR improvement for each number of bootstrapped mixtures which are considered under expectation $\frac{1}{B}\sum_{b=1}^{B} f_{\mathcal{S}}^{(\widehat{s})}(\widetilde{s}+\widetilde{n}'_b;\boldsymbol{\theta}_{\mathcal{S}})$. We show that as a fixed student network sees more input bootstrapped mixtures, the mean student performance becomes better on average than its teacher even early in training and in regions where the teacher performs poorly.

on multiple bootstrapped mixtures $\widetilde{m}'_b = \widetilde{s} + \widetilde{n}'_b$, $\forall b \in \{1, \dots, B\}$ produced using the same teacher's speech estimate $\widetilde{s}$ and independent teacher's noise estimates $\widetilde{n}'_b$ (see Equations 10, 11). The distribution of the SNR performance improvement that the empirical mean student yields over the initial teacher after 10 training epochs is displayed in Figure 5 while sweeping the number of input bootstrapped mixtures. For both cases of supervised and MixIT teachers we see that the mean SNR improvement is around 2 dB when increasing the number of bootstrapped mixtures B from 1 to 64. Notably, this result holds for really bad teacher estimates, namely, less than 5 dB and is obtained by simply performing inference over more augmentations of $\widetilde{s}$ without refining the student parameters. Assuming that all speech estimates and the ground-truth signals have unit-norm $\|s\| = \|\widetilde{s}\| = \|\widehat{s}_b\| = 1$, the maximization of SNR becomes equivalent to minimizing the $l2$ norm $\mathrm{SNR}(\widehat{y}, y) \propto -\|\widehat{y} - y\|$. As a result, the mean SNR improvement of the empirical mean student leads the term $\mathbb{E}[\langle \widehat{\mathbf{R}}_{\mathcal{S}}, \widetilde{\mathbf{R}}_{\mathcal{T}} \rangle]$ closer to zero (Equation 11) and consequently, the student to learn in a more robust way, even in cases where the teacher's error term $\|\widetilde{\mathbf{R}}_{\mathcal{T}}\|$ is far from zero.

F. Cross-domain generalization

A comparison for cross-domain generalization in self-supervised and semi-supervised domain adaptation speech enhancement tasks is displayed in Tables II and III, respectively.

In Table II, we notice that MixIT and its variants fail to generalize in cases where the noise distribution $\mathcal{D}_n$ does not closely resemble the true in-domain distribution $\mathcal{D}_n^*$. Notably, *RemixIT* outperforms all MixIT methods without having access to in-domain datasets. For instance, in the case where one only has access to mixtures from the WHAM! (W!) dataset and noise sources from LFSD (L), the best noise augmented MixIT model obtains only 1.6 dB of SI-SDR improvement on the adaptation WHAM! dataset. In stark contrast, *RemixIT* with a pre-trained MixIT teacher on LFSD yields an improvement of 5.3dB (1.6 $\rightarrow$ 6.9) over the best cross-dataset trained MixIT model and 0.7 dB (6.2 $\rightarrow$ 6.9) over the teacher model.

In very harsh mismatched cases, such as when using noise samples from LFSD and mixture samples from LFSD and WHAM!, *RemixIT* shows strong results for all datasets (4.9 dB for DNS, 7.2 dB for LFSD and 6.9 dB for WHAM!) while even the best MixIT configuration fails to produce significant improvements over the input mixture (1.7 dB for DNS, -1.7 dB for LFSD and 1.6 dB for WHAM!). Moreover, *RemixIT* can also improve the performance of a teacher model in the source dataset test-set by leveraging other target mixture datasets. Notice that *RemixIT* yields an improvement of 1.4dB (8.5 $\rightarrow$ 9.9) on the LFSD test-set over the pre-trained teacher model on LFSD by using self-training over the diverse unsupervised DNS mixture dataset. Surprisingly, *RemixIT* also outperforms its teacher by a large margin (6.2 $\rightarrow$ 8.2 dB) on the WHAM! test set even though it has not seen any data from this dataset which shows how *RemixIT* can provide a seamless solution to generalizing denoising models to unseen data.

Although *RemixIT* shows a small performance degradation in the adaptation $L \rightarrow$ **W!** set compared to the adaptation with cleaner datasets, such as: $L \rightarrow$ **D** (7.5 $\rightarrow$ 6.8 for the

Method	U	Training Data		Mean test-set		
		Noise	Mixtures	SI-SDRi (dB)		
		$\mathcal{D}_n$	$\mathcal{D}_m$	D	L	W!
MixIT with in-domain noise	8	D	D	5.2	6.3	6.6
	8^L	L	L	5.6	8.5	6.2
	8	W!	W!	4.5	2.3	**9.8**
MixIT with OOD noise	8	L	D	5.1	3.6	5.3
		W!		1.2	-0.2	1.7
		D	W!	-0.8	-6.3	1.8
		L		1.7	-1.7	1.2
MixIT with extra OOD injected noise [31]	8	L+L	D	5.3	1.3	4.7
		W!+W!		5.3	5.0	2.9
		D+D	W!	-3.5	-9.8	3.0
		L+L		-1.6	-10.2	1.6
RemixIT (ours)	8	L	L→**D**	6.3	9.3	7.5
	32			**6.8**	**9.9**	8.2
	8	L	L→**W!**	5.3	7.5	6.8
	32			4.9	7.2	6.9

TABLE II: Self-supervised training mean SI-SDR improvement (dB) over the input mixture performance for MixIT baselines and *RemixIT*. The initial MixIT teacher uses a sudo rm -rf [60] architecture with $U = 8$ blocks on the LFSD (L) dataset and is denoted with ^{L}. The evaluated *RemixIT*'s student models follow a sequentially updated teacher protocol where they grow in depth as: $U : 8 \rightarrow 16 \rightarrow 32$ and are only trained using the corresponding **bolded** mixture dataset. Gray background colored cells denote the best performing model which did not have access to clean in-domain training data for the corresponding dataset (note that MixIT assumes access to clean in-domain noise recordings). The mean noisy input mixture SI-SDR performance is 9.2, 6.3 and -1.3 dB for DNS, LFSD, and WHAM! datasets, respectively.

Method	U	Training Data			Mean test-set		
		Speech	Noise	Mixtures	SI-SDRi (dB)		
		$\mathcal{D}_s$	$\mathcal{D}_n$	$\mathcal{D}_m$	D	L	W!
Super-vised	8^D	D	D		9.4	10.3	8.4
	32				**10.5**	**12.2**	10.2
	$8^{W!}$	W!	W!		6.1	4.8	11.3
	32				7.1	6.2	**12.8**
RemixIT (ours)	8	D	D	**W!**	6.9	5.6	9.0
	32				6.9	5.4	9.8
	8	W!	W!	**D**	6.7	5.7	11.3
	32				7.3	6.4	10.9

TABLE III: *RemixIT* mean SI-SDR improvement (dB) over the input mixture for semi-supervised domain adaptation. The initial OOD supervised teachers with a sudo rm -rf [60] with $U = 8$ blocks on the DNS (D) and the WHAM! (W!) datasets are denoted with ^{D} and $^{W!}$, respectively. The evaluated *RemixIT*'s student models follow a sequentially updated teacher protocol where they grow in depth as: $U : 8 \rightarrow 16 \rightarrow 32$ and are only trained using the corresponding **bolded** mixture dataset. Gray background colored cells denote the best performing model which did not have access to clean in-domain training data for the corresponding dataset. The mean noisy input mixture SI-SDR performance is 9.2, 6.3 and -1.3 dB for DNS (D), LFSD (L), and WHAM! (W!) datasets, respectively.

shallow student and $8.2 \rightarrow 6.9$ for the $U = 32$ student on W!), notice that the same MixIT configuration suffers a major hit in denoising performance ($5.3 \rightarrow 1.2$ dB on W!) which makes it almost similar to a no-processing model. In essence, the input SNR of WHAM! (-1.3dB) prevents self-supervised algorithms from learning effectively and training on OOD but higher input-SNR datasets (e.g. DNS) leads to better results.

In Table III, we show that *RemixIT* aptly performs semi-supervised domain adaptation even for severely mismatched cases such as transferring knowledge from DNS to the much less diverse and lower input-SNR WHAM!. *RemixIT* yields the best performing model without clean in-domain source signals on the DNS test set (7.3 dB) when only starting from the OOD semi-supervised teacher on WHAM! with a much inferior performance of 6.1 dB.

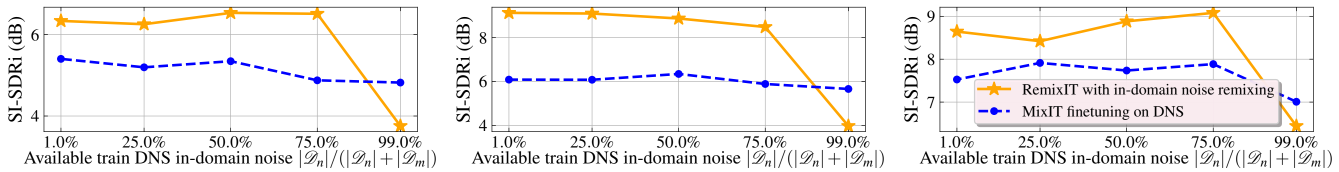
G. RemixIT with in-domain noise recordings

Finally, we also propose an extension to our proposed self-training method to adopt readily available isolated in-domain noise recordings $\mathbf{n} \sim \mathcal{D}_n$ which can further enhance *RemixIT*'s performance. To do so, we alter the bootstrapped remixing process presented in Equation 5 using a portion of the in-domain noise recordings $\mathbf{n} \sim \mathcal{D}_n$ instead of the teacher's noise estimates $\widetilde{\mathbf{n}} \sim f_{\mathcal{T}}^{\widetilde{\mathbf{n}}}(\mathbf{m}; \boldsymbol{\theta}_{\mathcal{T}}^{(k)})$ as shown below:

$$\widetilde{\mathbf{m}}_b = \widetilde{\mathbf{s}}_b + \zeta \mathbf{n}_b + (1-\zeta)\widetilde{\mathbf{n}}_b^{(\mathbf{P})}, \;\; \zeta \sim \text{Bernoulli}(p_n), \tag{17}$$

where b indicates the batch-index and p_n is the Bernoulli parameter of sampling an in-domain noise recording instead of a teacher's estimate for the corresponding batch-index.

In Figure 6 we show how our method performs against a stronger fine-tuned MixIT baseline using the same Sudo rm -rf architecture with $U = 8$ U-ConvBlocks on various splits of the DNS training data. The pre-trained model on LFSD data is used as an initialization checkpoint for MixIT fine-tuning and as the teacher network for performing *RemixIT* with bootstrapped mixtures from teacher's estimates and in-domain noise recordings. We set the probability of synthesizing a bootstrapped mixture with an isolated in-domain noise recording instead of a teacher's noise estimate equal to the ratio of the in-domain noise recordings compared to the mixture data $p_n = {}^{|\mathcal{D}_n|}/_{(|\mathcal{D}_n|+|\mathcal{D}_m|)}$. We notice that *RemixIT* performs consistently better than the fine-tuned MixIT for the same ratio of in-domain noise recordings except of the rightmost point where the bootstrapped mixtures contain less diverse mixtures leading the student model to overfit to only a small amount of human utterances. Notably, *RemixIT* trains a full student model from scratch compared to the fine-tuned MixIT which has more trainable parameters (0.97 millions vs 0.56) and also enjoys the warm-start from a LFSD MixIT checkpoint. It is also evident that our proposed *RemixIT* extension becomes better with more supervised data for the generalization datasets (see Figure 6a for DNS and Figure 6c for WHAM!). This is also reflected on a small ablation study that we performed to set the in-domain noise sampling prior parameter p_n in which we kept the amount of in-domain noise recordings and mixture data equal $|\mathcal{D}_n| = |\mathcal{D}_m|$ and gradually increased the Bernoulli parameter $p_n : 0.01 \rightarrow 0.5$. As a result, we noticed a performance increase in terms of SI-SDRi of $6.1 \rightarrow 6.4$ (dB) for the DNS test-set and $8.6 \rightarrow 9.0$ (dB) for the WHAM! dataset which enhances our claim that cleaner noise estimates



(a) Performance on the DNS test set.

(b) Performance on the LFSD test set.

(c) Performance on the WHAM! test set.

Fig. 6: SI-SDR (dB) performance improvement of a sudo rm -rf ($U = 8$) model fine-tuned using MixIT (blue-dashed line) and trained using *RemixIT* with in-domain noise recordings recordings remixing (solid orange line) on different test sets with different DNS training set splits. For each plot the x-axis denote the split between the DNS-training partition between in-domain noise recordings $\mathcal{D}_n$ and mixture available data $\mathcal{D}_m$. Both self-supervised speech enhancement methods start using the same pre-trained MixIT sudo rm -rf ($U = 8$) model with 20% in-domain isolated noise data and 80% mixture recordings from the LFSD dataset. For each method we evaluate the corresponding checkpoints that lead to the best performance on the LFSD test set after 20 full training epochs.

can lead to stronger gains through synthesizing bootstrapped mixtures with less interference.

V. Conclusion

We have presented a self-training scheme for speech enhancement models which is based on a lifelong bi-directional parameter update between a teacher and a student network. The proposed framework aptly transfers the knowledge of a pre-trained model on out-of-domain data using bootstrapped remixing and through the continual refinement of the teacher's outputs. We have experimentally shown that our method significantly outperforms all previous state-of-the-art self-supervised methods while being more general and without the dependence on in-domain data. Moreover, our results illustrated that *RemixIT* can also perform semi-supervised and zero-shot domain adaptation setups with limited in-domain mixtures. Furthermore, our theoretical analysis is backed by empirical results and instrumental to the understanding of the teacher-student learning dynamics, especially in where our method can still learn with extremely noisy pseudo-target signals. In the future, we aim to strengthen the robustness of our algorithm by estimating a confidence-based proxy for the quality of the pseudo-targets [39] as well as widen the applicability of our method by applying it to different domains.

References

[1] Jacob Benesty, Shoji Makino, and Jingdong Chen, *Speech enhancement*, Springer Science & Business Media, 2006.

[2] Felix Weninger, Hakan Erdogan, Shinji Watanabe, Emmanuel Vincent, Jonathan Le Roux, John R Hershey, and Björn Schuller, "Speech enhancement with lstm recurrent neural networks and its application to noise-robust asr," in *Latent Variable Analysis and Signal Separation*. Springer, 2015, pp. 91–99.

[3] Tara N Sainath, Ron J Weiss, Kevin W Wilson, Bo Li, Arun Narayanan, Ehsan Variani, Michiel Bacchiani, Izhak Shafran, Andrew Senior, Kean Chin, et al., "Multichannel signal processing with deep neural networks for automatic speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 5, pp. 965–979, 2017.

[4] Hassan Taherian, Zhong-Qiu Wang, Jorge Chang, and DeLiang Wang, "Robust speaker recognition based on single-channel and multi-channel speech enhancement," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1293–1302, 2020.

[5] Hyeong-Seok Choi, Jang-Hyun Kim, Jaesung Huh, Adrian Kim, Jung-Woo Ha, and Kyogu Lee, "Phase-aware speech enhancement with deep complex u-net," in *Proc. ICLR*, 2018.

[6] Yan Zhao, Zhong-Qiu Wang, and DeLiang Wang, "Two-stage deep learning for noisy-reverberant speech enhancement," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 27, no. 1, pp. 53–62, 2018.

[7] Alexandre Défossez, Gabriel Synnaeve, and Yossi Adi, "Real time speech enhancement in the waveform domain," in *Proc. Interspeech 2020*, 2020, pp. 3291–3295.

[8] Xiang Hao, Xiangdong Su, Radu Horaud, and Xiaofei Li, "Fullsubnet: A full-band and sub-band fusion model for real-time single-channel speech enhancement," in *Proc. ICASSP*, 2021, pp. 6633–6637.

[9] Han Zhao, Shuayb Zarar, Ivan Tashev, and Chin-Hui Lee, "Convolutional-recurrent neural networks for speech enhancement," in *Proc. ICASSP*, 2018, pp. 2401–2405.

[10] Yuma Koizumi, Kohei Yatabe, Marc Delcroix, Yoshiki Masuyama, and Daiki Takeuchi, "Speech enhancement using self-adaptation and multi-head self-attention," in *Proc. ICASSP*, 2020, pp. 181–185.

[11] Umut Isik, Ritwik Giri, Neerad Phansalkar, Jean-Marc Valin, Karim Helwani, and Arvindh Krishnaswamy, "Poconet: Better speech enhancement with frequency-positional embeddings, semi-supervised conversational data, and biased loss," in *Proc. Interspeech*, 2020, pp. 2487–2491.

[12] Ashutosh Pandey and DeLiang Wang, "Dense cnn with self-attention for time-domain speech enhancement," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 1270–1279, 2021.

[13] Santiago Pascual, Antonio Bonafonte, and Joan Serrà, "Segan: Speech enhancement generative adversarial network," in *Proc. Interspeech*, 2017, pp. 3642–3646.

[14] Szu-Wei Fu, Chien-Feng Liao, Yu Tsao, and Shou-De Lin, "Metricgan: Generative adversarial networks based black-box metric scores optimization for speech enhancement," in *Proc. ICML*, 2019, pp. 2031–2041.

[15] Yoshiaki Bando, Masato Mimura, Katsutoshi Itoyama, Kazuyoshi Yoshii, and Tatsuya Kawahara, "Statistical speech enhancement based on probabilistic integration of variational autoencoder and non-negative matrix factorization," in *Proc. ICASSP*, 2018, pp. 716–720.

[16] Yangyang Xia, Buye Xu, and Anurag Kumar, "Incorporating real-world noisy speech in neural-network-based speech enhancement systems," *arXiv preprint arXiv:2109.05172*, 2021.

[17] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Proc. NeurIPS*, 2020, vol. 33, pp. 12449–12460.

[18] Weiran Wang, Qingming Tang, and Karen Livescu, "Unsupervised pre-training of bidirectional speech encoders via masked reconstruction," in *Proc. ICASSP*, 2020, pp. 6889–6893.

[19] Andy T Liu, Shang-Wen Li, and Hung-yi Lee, "Tera: Self-supervised learning of transformer encoder representation for speech," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 2351–2366, 2021.

[20] Chien-Feng Liao, Yu Tsao, Hung-Yi Lee, and Hsin-Min Wang, "Noise Adaptive Speech Enhancement Using Domain Adversarial Training," in *Proc. Interspeech*, 2019, pp. 3148–3152.

[21] Ryandhimas E Zezario, Tassadaq Hussain, Xugang Lu, Hsin-Min Wang, and Yu Tsao, "Self-supervised denoising autoencoder with linear regression decoder for speech enhancement," in *Proc. ICASSP*, 2020, pp. 6669–6673.

[22] Aswin Sivaraman, Sunwoo Kim, and Minje Kim, "Personalized speech enhancement through self-supervised data augmentation and purification," in *Proc. Interspeech*, 2021, pp. 2676–2680.

[23] Aswin Shanmugam Subramanian, Xiaofei Wang, Murali Karthick Baskar, Shinji Watanabe, Toru Taniguchi, Dung Tran, and Yuya Fujita, "Speech enhancement using end-to-end speech recognition objectives," in *Proc. WASPAA*, 2019, pp. 234–238.

[24] Viet Anh Trinh and Sebastian Braun, "Unsupervised speech enhancement with speech recognition embedding and disentanglement losses," in *Proc. ICASSP*, 2022, pp. 391–395.

[25] Ying Cheng, Mengyu He, Jiashuo Yu, and Rui Feng, "Improving multimodal speech enhancement by incorporating self-supervised and curriculum learning," in *Proc. ICASSP*, 2021, pp. 4285–4289.

[26] Xiaofei Li and Radu Horaud, "Multichannel speech enhancement based on time-frequency masking using subband long short-term memory," in *Proc. WASPAA*, 2019, pp. 298–302.

[27] Kazuki Shimada, Yoshiaki Bando, Masato Mimura, Katsutoshi Itoyama, Kazuyoshi Yoshii, and Tatsuya Kawahara, "Unsupervised speech enhancement based on multichannel nmf-informed beamforming for noise-robust automatic speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 5, pp. 960–971, 2019.

[28] Scott Wisdom, Efthymios Tzinis, Hakan Erdogan, Ron J Weiss, Kevin Wilson, and John R Hershey, "Unsupervised sound separation using mixture invariant training," in *Proc. NeurIPS*, 2020, vol. 33, pp. 3846–3857.

[29] Takuya Fujimura, Yuma Koizumi, Kohei Yatabe, and Ryoichi Miyazaki, "Noisy-target training: A training strategy for dnn-based speech enhancement without clean speech," in *Proc. EUSIPCO*, 2021, pp. 436–440.

[30] Efthymios Tzinis, Jonah Casebeer, Zhepei Wang, and Paris Smaragdis, "Separate but together: Unsupervised federated learning for speech enhancement from non-iid data," in *Proc. WASPAA*, 2021, pp. 46–50.

[31] Koichi Saito, Stefan Uhlich, Giorgio Fabbro, and Yuki Mitsufuji, "Training speech enhancement systems with noisy speech datasets," *arXiv preprint arXiv:2105.12315*, 2021.

[32] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz, "Mixup: Beyond empirical risk minimization," in *Proc. ICLR*, 2018.

[33] Ryo Aihara, Toshiyuki Hanazawa, Yohei Okato, Gordon Wichern, and Jonathan Le Roux, "Teacher-student deep clustering for low-delay single channel speech separation," in *Proc. ICASSP*, 2019, pp. 690–694.

[34] Wei-Ning Hsu, Yao-Hung Hubert Tsai, Benjamin Bolte, Ruslan Salakhutdinov, and Abdelrahman Mohamed, "Hubert: How much can a bad teacher benefit asr pre-training?," in *Proc. ICASSP*, 2021, pp. 6533–6537.

[35] Jisi Zhang, Cătălin Zorilă, Rama Doddipatla, and Jon Barker, "Teacher-Student MixIT for Unsupervised and Semi-Supervised Speech Separation," in *Proc. Interspeech*, 2021, pp. 3495–3499.

[36] Sunwoo Kim and Minje Kim, "Test-time adaptation toward personalized speech enhancement: Zero-shot learning with knowledge distillation," in *Proc. WASPAA*, 2021, pp. 176–180.

[37] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le, "Unsupervised data augmentation for consistency training," *Proc. NeurIPS*, vol. 33, pp. 6256–6268, 2020.

[38] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel, "Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring," in *Proc. ICLR*, 2019.

[39] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," in *Proc. NeurIPS*, 2020, vol. 33, pp. 596–608.

[40] Zhepei Wang, Ritwik Giri, Umut Isik, Jean-Marc Valin, and Arvindh Krishnaswamy, "Semi-supervised singing voice separation with noisy self-training," in *Proc. ICASSP*, 2021, pp. 31–35.

[41] Antti Tarvainen and Harri Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Proc. NeurIPS*, 2017, vol. 30, pp. 1195–1204.

[42] Yosuke Higuchi, Niko Moritz, Jonathan Le Roux, and Takaaki Hori, "Momentum pseudo-labeling for semi-supervised speech recognition," in *Proc. Interspeech*, 2021, pp. 726–730.

[43] Qiantong Xu et al., "Iterative pseudo-labeling for speech recognition," in *Proc. Interspeech*, 2020, pp. 1006–1010.

[44] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le, "Self-training with noisy student improves imagenet classification," in *Proc. CVPR*, 2020, pp. 10684–10695.

[45] Scott Wisdom, John R Hershey, Kevin Wilson, Jeremy Thorpe, Michael Chinen, Brian Patton, and Rif A Saurous, "Differentiable consistency constraints for improved deep speech enhancement," in *Proc. ICASSP*, 2019, pp. 900–904.

[46] Dong Yu, Morten Kolbæk, Zheng-Hua Tan, and Jesper Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *Proc. ICASSP*, 2017, pp. 241–245.

[47] Midia Yousefi, Soheil Khorram, and John H.L. Hansen, "Probabilistic Permutation Invariant Training for Speech Separation," in *Proc. Interspeech*, 2019, pp. 4604–4608.

[48] Efthymios Tzinis, Shrikant Venkataramani, Zhepei Wang, Cem Subakan, and Paris Smaragdis, "Two-step sound source separation: Training on learned latent targets," in *Proc. ICASSP*, 2020, pp. 31–35.

[49] Michael Fleermann and Werner Kirsch, "The central limit theorem for weakly dependent random variables by the moment method," *arXiv preprint arXiv:2202.04717*, 2022.

[50] Chandan KA Reddy et al., "The interspeech 2020 deep noise suppression challenge: Datasets, subjective testing framework, and challenge results," in *Proc. Interspeech*, 2020, pp. 2492–2496.

[51] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *Proc. ICASSP*, 2015, pp. 5206–5210.

[52] Eduardo Fonseca, Xavier Favory, Jordi Pons, Frederic Font, and Xavier Serra, "Fsd50k: an open dataset of human-labeled sound events," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 829–852, 2021.

[53] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. ICASSP*, 2017, pp. 776–780.

[54] Gordon Wichern, Joe Antognini, Michael Flynn, Licheng Richard Zhu, Emmett McQuinn, Dwight Crow, Ethan Manilow, and Jonathan Le Roux, "WHAM!: Extending Speech Separation to Noisy Environments," in *Proc. Interspeech*, 2019, pp. 1368–1372.

[55] Ritwik Giri, Umut Isik, and Arvindh Krishnaswamy, "Attention wave-u-net for speech enhancement," in *Proc. WASPAA*, 2019, pp. 249–253.

[56] Junichi Yamagishi, Christophe Veaux, and Kirsten MacDonald, "CSTR VCTK Corpus: English multi-speaker corpus for CSTR voice cloning toolkit (version 0.92)," 2019.

[57] Joachim Thiemann, Nobutaka Ito, and Emmanuel Vincent, "The diverse environments multi-channel acoustic noise database: A database of multichannel environmental noise recordings," *The Journal of the Acoustical Society of America*, vol. 133, no. 5, pp. 3591–3591, 2013.

[58] Efthymios Tzinis, Zhepei Wang, and Paris Smaragdis, "Sudo rm-rf: Efficient networks for universal audio source separation," in *Proc. MLSP*, 2020, pp. 1–6.

[59] Yi Luo, Cong Han, and Nima Mesgarani, "Ultra-lightweight speech separation via group communication," in *Proc. ICASSP*, 2021, pp. 16–20.

[60] Efthymios Tzinis, Zhepei Wang, Xilin Jiang, and Paris Smaragdis, "Compute and memory efficient universal sound source separation," *Journal of Signal Processing Systems*, vol. 94, no. 2, pp. 245–259, 2022.

[61] Jonathan Le Roux, Scott Wisdom, Hakan Erdogan, and John R Hershey, "Sdr–half-baked or well done?," in *Proc. ICASSP*, 2019, pp. 626–630.

[62] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[63] Neil Zeghidour and David 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.

[64] Cem Subakan, Mirco Ravanelli, Samuele Cornell, Mirko Bronzi, and Jianyuan Zhong, "Attention is all you need in speech separation," in *Proc. ICASSP*, 2021, pp. 21–25.

[65] Cees H Taal, Richard C Hendriks, Richard Heusdens, and Jesper Jensen, "An algorithm for intelligibility prediction of time–frequency weighted noisy speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2125–2136, 2011.

[66] Antony W Rix, John G Beerends, Michael P Hollier, and Andries P Hekstra, "Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs," in *Proc. ICASSP*, 2001, pp. 749–752.

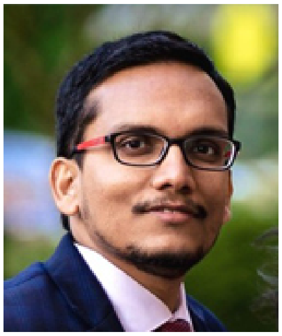
Efthymios Tzinis is a PhD candidate in the Computer Science department at the University of Illinois Urbana-Champaign (UIUC). He also holds a diploma (BSc and MEng) in Electrical and Computer Engineering (ECE) from the National Technical University of Athens (NTUA). Efthymios has previously conducted research for Google Research, Mitsubishi Electric Research Laboratories (MERL) as well as Reality Labs Research, Meta. His research interests lie in unsupervised audio source separation, audio-visual perception and signal processing, efficient neural networks, as well as federated learning under real-world non-IID settings. He is also the recipient of the Google PhD fellowship.

Yossi Adi is an Assistant Professor at the Hebrew University of Jerusalem and Research Scientist at Meta AI Research. Yossi completed his Ph.D. in computer science at Bar-Ilan University. Yossi's research interests are in speech and language processing using machine learning and deep learning models. Yossi's research spans core machine learning and deep learning algorithms, their applications to spoken language processing, and the impact of the technology on social systems.

Vamsi K Ithapu leads is a Research Science Manager and Technical Lead at Meta Reality Labs (RL). He leads the AI group at Meta RL Research Audio. He finished his PhD from University of Wisconsin Madison in 2018 and prior to that he obtained his Bachelors in Technology from Indian Institute of Technology Guwahati in 2010. His research interests lie at the intersection of machine learning, computer vision and augmented/virtual reality (ARVR), specifically with applications in audio-driven multi-sensory egocentric experiences. He is a member of IEEE and has over 11 patents in ARVR.

Buye Xu received the B.S. (2000) in Electrical Engineering from Nanjing University, Nanjing, China, and the Ph.D. (2010) in Physics from Brigham Young University, Provo, UT, U.S.A.. He was a DSP research engineer in Starkey Hearing Technologies, and is currently a research scientist in Meta Reality Labs Research. His research has centered on the areas related to personal hearing devices, including speech processing, auditory perception, hearing impairment, active noise control, etc. He served as Associate Editor of JASA Express Letters.

Paris Smaragdis is a professor of Computer Science in the University of Illinois at Urbana-Champaign. He competed his graduate studies at MIT, and before coming to UIUC was a research scientist at MERL and Adobe. His research is focused on machine learning approaches to solving various audio signal processing problems. In 2006 he was selected by MIT's Technology Review as one of the year's top young technology innovators (TR35) for his work on machine listening, he has won the IEEE Signal Processing Society Best Paper Award twice (2017 and 2020), he was elected an IEEE Fellow (class of 2015), and selected as an IEEE Signal Processing Society Distinguished Lecturer (2016-2017). He has served in the IEEE Signal Processing Society Board of Governors (2017-2020), as chair of the Machine Learning for Signal Processing Technical Committee of the IEEE (2012-2014), as chair of the Audio and Acoustics Signal Processing Technical Committee of the IEEE (2018-2020), as chair of the IEEE SPS Data Science Initiative (2019-2020), and as chair of the Audio and Acoustics Signal Processing Technical Committee of the IEEE. He has been a Senior Area Editor for IEEE's Signal Processing Transactions and IEEE's Open Journal of Signal Processing, and is currently the Editor-in-Chief of the the IEEE/ACM Transactions in Audio, Speech and Language Processing. His research has been productized multiple times in commercial software that is in use by millions of users worldwide, he holds more than 40 US patents as well as patents in Japan and Europe. He has been an active consultant on audio technologies with multiple Fortune 500 companies.



Anurag Kumar (Member, IEEE) is currently a Research Scientist Reality Labs Research, Meta. Before joining Meta, Anurag obtained his PhD from Language Technologies Institute (LTI) in School of Computer Science, Carnegie Mellon University in 2018 and has been at Meta Reality Labs Research since then. Anurag obtained his undergraduate degree in Electrical Engineering from IIT Kanpur in 2013. Anurag's primary research interests are deep learning, machine learning and audio and speech processing.