Efficient Personalized Speech Enhancement through Self-Supervised Learning

Aswin Sivaraman, *Student Member, IEEE*, Minje Kim, *Senior Member, IEEE*

***Abstract*—This work presents self-supervised learning methods for developing monaural speaker-specific (i.e., personalized) speech enhancement models. While generalist models must broadly address many speakers, specialist models can adapt their enhancement function towards a particular speaker's voice, expecting to solve a narrower problem. Hence, specialists are capable of achieving more optimal performance in addition to reducing computational complexity. However, naive personalization methods can require clean speech from the target user, which is inconvenient to acquire, e.g., due to subpar recording conditions. To this end, we pose personalization as either a zero-shot task, in which no additional clean speech of the target speaker is used for training, or a few-shot learning task, in which the goal is to minimize the duration of the clean speech used for transfer learning. With this paper, we propose self-supervised learning methods as a solution to both zero- and few-shot personalization tasks. The proposed methods are designed to learn the personalized speech features from unlabeled data (i.e., in-the-wild noisy recordings from the target user) without knowing the corresponding clean sources. Our experiments investigate three different self-supervised learning mechanisms. We set up a pseudo speech enhancement problem as a pretext task, which pretrains the models to estimate noisy speech as if it were the clean target. Contrastive learning and data purification methods regularize the loss function of the pseudo enhancement problem, overcoming the limitations of learning from unlabeled data. We assess our methods by personalizing the well-known ConvTasNet architecture to twenty different target speakers. The results show that self-supervised models achieve zero-shot and few-shot personalization using fewer model parameters and less clean data from the target user, achieving the data efficiency and model compression goals.**

***Index Terms*—Personalized speech enhancement, self-supervised learning, data efficiency, model complexity**

I. Introduction

WITH the ubiquity of voice-controlled intelligent devices, there is now an ever-growing demand for low-cost robust speech processing systems. These systems are reliant on speech enhancement (SE) technology, which improves the quality and intelligibility of noisy speech signals [1]. Over the last decade, deep learning algorithms have quickly defined the state-of-the-art in SE research [2]–[10]. Most neural networks proposed for SE are trained using supervised learning frameworks [11]. Typically, input-output pairs are programmatically generated by mixing various speech recordings with assorted noise recordings. Supervised data preparation requires labeled datasets, i.e., the speech signals are known to be clean or of reference quality. The neural networks then learn a mapping function between the input mixture signals and their originating ground-truth clean speech signal. Consequently, the learning outcomes of supervised SE models are highly dependent on the diversity of the training data and increased model complexity. But because large publicly available datasets do not represent all populations, machine learning models can easily become biased towards over-represented social groups [12]. For instance, a general-purpose universal SE model may under-perform for a particular target speaker if their unique vocal characteristics or noisy environment are never encountered during training. And although modern GPU technology enables training neural networks with millions of learnable parameters [13], bigger models are infeasible to deploy on low-resource devices [14]. To overcome these disadvantages of generalist models, we focus our investigation on developing speaker-specific specialist models. Through model specialization, we narrow down the scope of the SE task, unlocking the potential for improved enhancement of a single target speaker while affording a reduction in model complexity. We refer to this narrowed problem definition as the personalized speech enhancement (PSE) task. Figure 1 illustrates this idea of replacing large generalist models with smaller more-optimized specialist models.

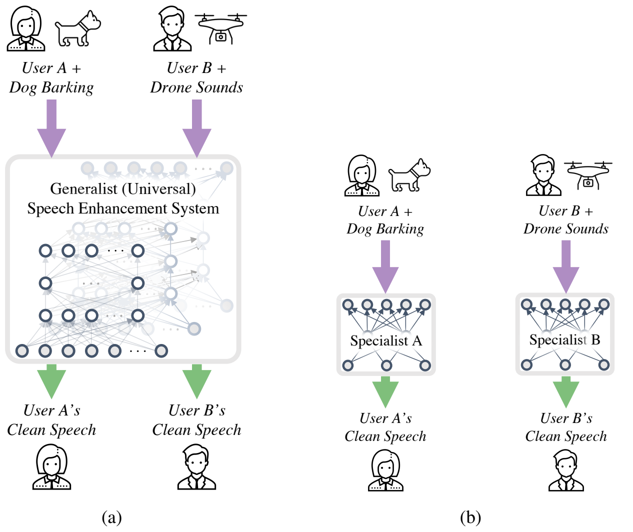


Fig. 1. A conceptual overview of how specialist models can replace generalist models for speech enhancement. Model inputs and outputs are shown in purple and green arrows, respectively. For example, Specialist A can train using User A's unique noisy speech recordings, producing a speaker-specific cleaned output. Optimizing for a specific speaker or environment can yield improved performance. Additionally, because specialists address a subset of the generalist's task, they can theoretically afford a reduction in model parameters.

While specialists are theoretically preferable to generalists, in practice, personalization is a challenging optimization strategy because it requires explicit knowledge about the target speaker and their environment. Realistically, users of voice-controlled devices would be reluctant to share their clean voice data due to privacy concerns. Given that speech synthesis models can be conditioned on speaker identity using only five seconds of clean speech data [15], unwanted forgery of one's voice is a genuine issue. Even if users are willing to provide their clean speech, it can still be difficult to collect a large quantity of studio-quality noise-free recordings. Users might not have access to a quiet anechoic room, and their microphones may introduce unwanted artifacts.

This study addresses the voice-controlled device PSE task considering the aforementioned user data constraints. We envision two possible scenarios. In one, the smart device must personalize using only the target speaker's in-the-wild noisy speech data. We view this as a *zero-shot* (ZSL) machine learning problem as there is no labeled data. In the second scenario, the target user provides the device with a limited amount of clean speech, on the order of seconds. If these signals are of good quality, we consider personalization to be a *few-shot* (FSL) problem, given that some labeled speech is now available. Even if there is abundant unlabeled data, the supervised training framework cannot be applied because the data is not guaranteed to be reference-quality.

In this paper, we assess two self-supervised learning (SSL) frameworks for developing PSE models. Through SSL, we train the PSE model to solve a "pretext task" using the unlabeled data. The effectiveness of the model's learned features in SSL depends on how mismatched the pretext task is from the intended downstream task. This paradigm has gained popularity in many research areas such as computer vision [16], natural language processing [17], and reinforcement learning [18]. One particular SSL technique, known as contrastive learning, augments the unlabelled data in a pairwise manner. Training a model to keep similar samples together becomes an easy method for developing discriminative feature spaces. As suggested in the recent SimCLR paper, the composition of the pairwise data augmentation pipeline strongly influences the robustness of the learned contrastive features [19]. Principally, self-supervision differs from traditional supervised learning by removing the necessity for labeled data. Instead, through SSL, the goal is to learn representative features that are useful for the downstream task. Consequently, our goal in this paper is to devise data augmentation techniques and contrastive loss functions that can help extract meaningful features from the unlabeled data—which in our case correspond to noisy speech.

In place of the unavailable clean speech references, both SSL procedures repurpose the target speaker's noisy speech as the new training target. The first method we investigate is *pseudo speech enhancement* (PseudoSE), referred to in other literature as *noisy target training*. We mix further noise onto the already-noisy observations and train the PSE model to remove the newly injected noise. This self-supervised task is mismatched from true speech enhancement since the input signals are doubly degraded. Therefore, the upper bound of PseudoSE is determined by how noisy the in-the-wild data originally is. Nevertheless, specialist models trained using PseudoSE have an advantage over generalists because their training data is *in-domain*, i.e., the noisy speech signals are still speaker-specific as opposed to speaker-agnostic.

The second method we investigate is *contrastive mixtures* (CM). Here, we reorganize the doubly-degraded input signals into pairs. We prepare positive pairs—which share the same noisy speech source but have different injection noises. Conversely, the negative pairs have identical injection noises but differing noisy speech sources. Once the paired inputs are pseudo-denoised, the PSE model must learn to maximize the similarity of positive pair outputs and minimize the similarity of negative pair outputs. CM improves upon PseudoSE thanks to the contrastive learning terms. Negative pairs, in particular, provide an additional learning opportunity. Note that speech enhancement can be seen as a subset of source separation, wherein the two sources are known to be speech and noise. Any additional utilization of the source separation nature of the problem will be a plus. Hence, with CM, when preparing negative pairs, we see the noise source as the shared primary source while the speech sources are considered interfering sources that disagree. Another view is that the negative pairs' disagreement objective acts as a regularization of PseudoSE.

Additionally, we show that both SSL frameworks can benefit from a *data purification* (DP) process. As its name suggests, DP makes the unlabeled noisy speech signals more useful by identifying the parts that contain purer speech than others. In that regard, DP can be seen as an "active learning" method [20] where the goal is to focus on more important samples from a large unlabeled dataset. Regarding speech enhancement, we utilize DP by training a neural network in advance to predict speech quality frame-by-frame. Prior research has shown the feasibility of predicting the long-term signal-to-noise ratio (SNR) of a noisy speech signal [21]. Our quality predictor, which estimates segmental SNR, identifies the relatively cleaner segments within all the noisy unlabelled data, a process similar to auto-labeling. We convert the quality predictor's estimates into weights which guide the personalized speech enhancement model's loss function to prioritize cleaner input frames [22]. Then, the PseudoSE function becomes refined through data purification: the frames of audio that are doubly degraded become diminished through weighting. Consequently, the PseudoSE function will resemble an ideal speaker-specific fully-supervised learning process where only a single noise injection remains.

In summary, this research study explores personalized speech enhancement (PSE) as either a zero-shot (ZSL) or few-shot (FSL) learning problem. We investigate two self-supervised (SSL) methods for training PSE models, and we augment both methods using data purification (DP). Through our experiments, we assess the efficiency of the training methods in terms of data and model complexity. Notably, in the FSL context, data efficiency is achieved by developing models which utilize as little user-provided clean speech as possible. To this end, our experiment results show that models pretrained using the proposed self-supervised methods see greater improvements using a smaller amount of clean speech as opposed to fully-supervised generalist pretrained models. Our experiments also assess PSE models of four different sizes, reinforcing the idea that in-domain self-supervised training allows smaller models to achieve a more competitive speech enhancement performance.

II. Related Works

We treat PSE as a data-driven problem, where training using in-domain data implicitly informs the model about the target speaker and their environment. This differs from other studies, which tackle model personalization using auxiliary metadata, e.g., speaker-identifying embedding vectors [23]–[25].

As mentioned in Sec. I, we posit that PSE models benefit over SE models for optimized performance but also for lossless model compression. Prior research has empirically shown that model specialization does lead to performance gains [26], [27]. Intuitively, focusing on a small subset of the initially complex problem simplifies the neural network learning objective. Therefore, with voice-controlled devices, we envision that a truly personalized speech enhancement model can optimally improve the experience of its primary user, forgoing other speakers and other unlikely acoustic scenarios.

Although model compression is an active area in deep learning research, many standardized methods, such as quantization or pruning [28], do not consider the context of the model after deployment. Decreasing the total number of model parameters without reformulating the model objective is an option, but this may result in discernible performance trade-offs [29]. Particularly with regards to speech enhancement or speech separation, more recent research has focused on novel model compression methods, including bitwise operations [30]–[33] or group communication in intermediate neural network layers [34]. These works successfully minimize the performance trade-off but miss the opportunity to exploit the model's deployment environment. With personalization, because the sub-problem is easier to solve, a compressed specialist model suffices to perform on par with a more complex generalist model. For example, using a model selection approach conditioned on speaker genders, a specialist using 512 hidden units produced speech signal improvement comparable to a generalist model using 1024 hidden units, yielding an effectively "lossless" 50 % reduction in run-time computational complexity [35].

Model selection is another approach for designing adaptive models. Four recent studies investigated run-time model selection as another means for test-time adaptation [35]–[38]. To develop a speech enhancement network through model selection, one must first cluster a large training corpus into non-overlapping subsets. Next, separate specialist submodules must be trained, each optimized around one of those subsets. Lastly, the network requires a classifier that assigns the unseen speaker's noisy utterances to the best-suited submodule at test-time. Sparse activation of these submodules helps to optimize performance and reduce run-time computational complexity [35]. By its design, model selection is inherently a zero-shot solution because the enhancement network does not utilize any prior knowledge about the target user. Rather, using a noise-robust classifier and choosing the best submodule proxies adaptation. We note that if all the submodules are very active during test-time, then the memory footprint savings of model selection are limited. Additionally, the network's spatial complexity linearly scales with the number of clusters, making model selection impractical for edge computing devices.

Another recent study explored knowledge distillation as a proxy for test-time personalization that can overcome all of the pitfalls mentioned above [39]. This dual-network configuration works by first training a teacher model, which processes test-time noisy signals from the unknown target speaker to produce pseudo-clean speech targets. A student model (with significantly fewer parameters than the teacher model) must learn using these pseudo targets. Through this framework, the student model is adapted to the target speaker's characteristics and target environment. We must recognize that the student model's performance is fundamentally upper-bounded by the teacher model's, favoring a larger and more powerful teacher. If the teacher model is prohibitively large, it must be placed off-device, which entails online-offline parameter updating procedures. Thus, in noting the limitations of model selection and knowledge distillation, we put forward resource-efficient methods for personalizing a model in this paper.

Other research in the last few years has explored SSL for general-purpose speech enhancement. An early work employed zero-shot SSL in a student-teacher framework, showing a student network that implicitly learned to perform speech enhancement despite being trained to minimize automatic speech recognition error [40]. More recently, another work describes an SSL framework that uses two autoencoders, trained to reproduce either clean speech or noisy speech [41]. The authors enforce a coupling of the two autoencoders' latent spaces through cycle-consistency loss functions. At inference time, the autoencoder trained only using mixture signals has its decoder swapped out, thus achieving zero-shot speech enhancement. These studies are limited to speaker-agnostic enhancement, and in particular, do not exploit self-supervised learning as a method for in-domain training. In contrast, two recent studies investigated using noisy speech data as target signals specifically for in-domain speech enhancement training [42], [43]. The PseudoSE method of this paper is similar to what they propose, however, our study investigates the benefits of noisy training targets specifically with regards to single-speaker model personalization and model compression. Additionally, our study is the first to bootstrap noisy target training with contrastive learning with regards to speech enhancement.

Recently, an SSL framework known as mixture invariant training (MixIT) [44] was proposed as an alternative to the fully-supervised permutation invariant training (PIT). It is a procedure for developing source separation systems using only mixtures of mixtures (MoM), i.e., linear combinations of arbitrary audio signals. Considering MixIT as a pretext task, it introduces systematic mismatch by design since the input MoMs have twice the number of expected sources at test-time. A recent study used MixIT by successfully adapting models to a set of speakers through joint training over in-domain and out-of-domain data [45], however the model compression implications were unexplored. In comparison to MixIT, the PseudoSE task may be viewed as a more speech enhancement-oriented version: while MixIT estimates every composite signal, PseudoSE learns explicitly from the combination of a target speaker's noisy utterance plus an injection noise. Therefore, a PseudoSE model is able to target the *pseudo* speech source and can omit reconstructing the injection noise.

III. Baseline Fully-Supervised Method for Personalized Speech Enhancement

For our discussion, we assume a hypothetical set $\mathbb{S}$ that encompasses all of the target speaker's clean utterances. Given the privacy concerns and technical difficulties mentioned in Sec. I, we assume that this set is inaccessible to the training algorithm; therefore, it cannot be used for personalization. In the FSL context, the short recordings provided by the target speaker represent a small subset of their unavailable ground-truth clean speech, i.e., $\mathbb{S}_{\textbf{f-tr}} \in \mathbb{S}$. The simplest approach for developing a personalized speech enhancement model would be to formulate a fully-supervised task over this subset. However, we theorize that the limited amount of data may result in suboptimal generalization performance and over-fitting. To remedy this issue, instead of randomly initializing the personalized model's parameters, one can first train a speaker-agnostic model and then finetune its parameters using $\mathbb{S}_{\textbf{f-tr}}$. Then, using transfer learning, we adapt a generalist model into a specialist model.

A. Training a Generalist

Training a generalist requires a large set of many anonymous speakers $\mathbb{G}$ as well as a large set of various non-stationary noises $\mathbb{N}$. A training set of artificial mixture signals $\boldsymbol{x}$ can be made by selecting random utterances $\boldsymbol{s} \in \mathbb{G}_{\text{tr}}$ and noises $\boldsymbol{n} \in \mathbb{N}_{\text{tr}}$ and summing the signals, i.e. $\boldsymbol{x} = \boldsymbol{s} + \boldsymbol{n}$. With each mixture, one may randomly scale $\boldsymbol{n}$ to be louder or quieter, thereby exposing the model to mixtures with varying signal-to-noise ratios (SNR). The generalist model can be described as a mapping function $f(\cdot)$ with parameters $\mathcal{W}_{\text{SE}}$ which is trained such that $f(\boldsymbol{x}; \mathcal{W}_{\text{SE}}) = \boldsymbol{y} \approx \boldsymbol{s}$, where the estimate $\boldsymbol{y}$ approximates the training target $\boldsymbol{s}$. The generalist's loss function $\mathcal{L}_{\text{SE}}$ is equivalent to the discrepancy between estimates and targets: $\mathcal{E}(\boldsymbol{y} \parallel \boldsymbol{s})$.

$$\mathcal{L}_{\text{SE}} = \mathcal{E}(\boldsymbol{y} \parallel \boldsymbol{s}) \tag{1}$$

$$\mathcal{W}_{\text{SE}} \leftarrow \underset{\mathcal{W}_{\text{SE}}}{\arg\min}\, \mathcal{L}_{\text{SE}} \tag{2}$$

There are many possible choices for the signal discrepancy function $\mathcal{E}$. The well-known signal-to-distortion ratio (SDR) metric [46] is frequently used as a general-purpose loss function for fully-supervised monaural time-domain speech enhancement [47]. A larger SDR correlates to improved speech quality, so when used as a neural network loss function, we minimize the negative of SDR. For a source signal $\boldsymbol{v}$ and estimate signal $\hat{\boldsymbol{v}}$, negative SDR loss is defined as follows:

$$\mathcal{E}_{\text{SDR}}(\hat{\boldsymbol{v}} \parallel \boldsymbol{v}) = -10 \log_{10} \left[\frac{\sum_t (v_t)^2}{\sum_t (v_t - \hat{v}_t)^2} \right]. \tag{3}$$

For generalists, what matters most is their generalization power. Although synthetic mixtures for fully-supervised training are straightforward to construct, models with low architectural complexity may not learn much from the data. That is, a smaller model may fail to enhance certain speakers' voices or remove particular noises—even if the training corpora for speech and noise signals were very large. In contrast, a bigger model may generalize very well, but using it in a resource-constrained device could be burdensome.

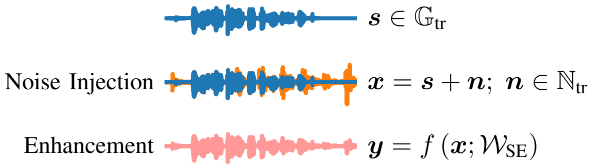

$\boldsymbol{s} \in \mathbb{G}_{\text{tr}}$
Noise Injection
$\boldsymbol{x} = \boldsymbol{s} + \boldsymbol{n};\ \boldsymbol{n} \in \mathbb{N}_{\text{tr}}$
Enhancement
$\boldsymbol{y} = f(\boldsymbol{x}; \mathcal{W}_{\text{SE}})$

Fig. 2. Multi-speaker (fully-supervised) speech enhancement setup. The training target is clean speech $\boldsymbol{s}$ and the model parameters $\mathcal{W}_{\text{SE}}$ are iteratively updated to minimize the loss function $\mathcal{L}_{\text{SE}}$. In the FSL context, we can finetune the model by sampling $\boldsymbol{s}$ from the small speaker-specific dataset $\mathbb{S}_{\textbf{f-tr}}$.

B. Personalization via Transfer Learning

The speaker-agnostic speech enhancement model may then be finetuned around the particular test-time speaker using transfer learning. Transfer learning is a straightforward fully-supervised approach to personalization, which handles the gap between the large multi-speaker dataset $\mathbb{G}$ and the small target speaker-provided clean dataset $\mathbb{S}_{\textbf{f-tr}}$. To do this, we create speaker-specific artificial mixture signals $\boldsymbol{x}$ composed stochastically by sampling from the limited subset $\boldsymbol{s} \in \mathbb{S}_{\textbf{f-tr}}$ and the training noises $\boldsymbol{n} \in \mathbb{N}_{\text{tr}}$. The parameters $\mathcal{W}_{\text{SE}}$ are once again iteratively updated in order to minimize the distance between estimate signals $\boldsymbol{y}$ and target signals $\boldsymbol{s}$. The finetuning loss function is equivalent to Eq. (2), but during finetuning, the model receives exposure to utterances from the target speaker.

The success of transfer learning as a personalization method depends on how effective the pretraining and finetuning steps are. For example, a large model highly generalized thanks to pretraining might barely adjust its parameters during finetuning. On the other hand, smaller models with weaker generalization capabilities may see a more significant performance boost through finetuning. Ultimately, the success of finetuning is primarily tied to the quality and quantity of the finetuning dataset $\mathbb{S}_{\textbf{f-tr}}$. Suppose the number of signals within $\mathbb{S}_{\textbf{f-tr}}$ is too few; in that case, finetuning may fail to improve performance even though $\mathbb{S}_{\textbf{f-tr}}$ consists of the target speaker's vocal characteristics. Also, because the FSL context only applies when the target speaker manually provides their clean speech, transfer learning is not viable without $\mathbb{S}_{\textbf{f-tr}}$.

Fig. 2 shows a visualization of the baseline pretraining process. The same signal transformations occur during transfer learning, when adapting the generalist model into a specialist model. If the target speaker does not provide $\mathbb{S}_{\textbf{f-tr}}$, the generalist model remains unadapted and therefore non-personalized.

IV. Proposed Self-Supervised Methods for Personalized Speech Enhancement

Here we describe our proposed self-supervised learning (SSL) methods, designed to improve the performance of the personalized speech enhancement models in either FSL or ZSL contexts. Through SSL, we aim at pretraining an SE model that can surpass the performance of the baseline generalist. This pretraining can suffice as a personalized solution (i.e., ZSL). Or, we can further finetune the self-supervised model by using the small amount of target speech signals if they are available (i.e., FSL).

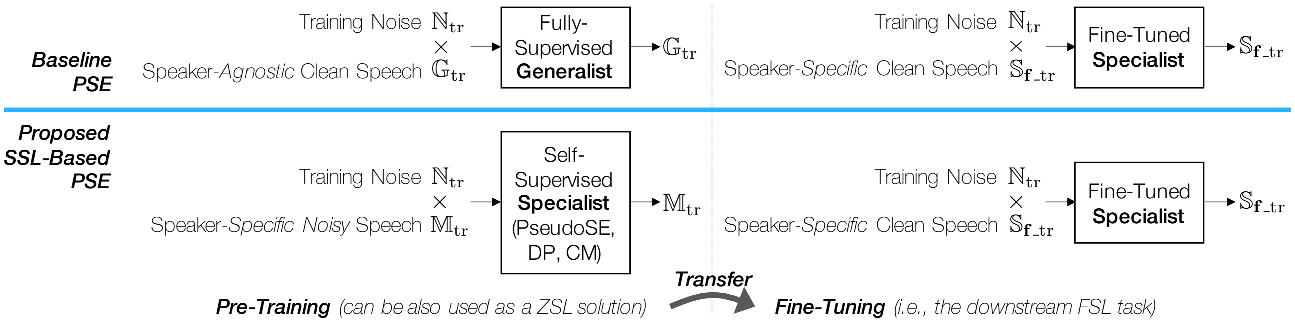

Baseline
PSE
Training Noise $\mathbb{N}_{\text{tr}}$
×
Speaker-*Agnostic* Clean Speech $\mathbb{G}_{\text{tr}}$
Fully-
Supervised
Generalist
$\mathbb{G}_{\text{tr}}$
Training Noise $\mathbb{N}_{\text{tr}}$
×
Speaker-*Specific* Clean Speech $\mathbb{S}_{\mathbf{f}\_\text{tr}}$
Fine-Tuned
Specialist
$\mathbb{S}_{\mathbf{f}\_\text{tr}}$
Proposed
SSL-Based
PSE
Training Noise $\mathbb{N}_{\text{tr}}$
×
Speaker-*Specific* *Noisy* Speech $\mathbb{M}_{\text{tr}}$
Self-
Supervised
Specialist
(PseudoSE,
DP, CM)
$\mathbb{M}_{\text{tr}}$
Training Noise $\mathbb{N}_{\text{tr}}$
×
Speaker-*Specific* Clean Speech $\mathbb{S}_{\mathbf{f}\_\text{tr}}$
Fine-Tuned
Specialist
$\mathbb{S}_{\mathbf{f}\_\text{tr}}$
Transfer
Pre-Training (can be also used as a ZSL solution)
Fine-Tuning (i.e., the downstream FSL task)

Fig. 3. An overview of the baseline and proposed personalization methods. With the baseline, the SE model is first pretrained using speaker-agnostic dataset as a generalist and then finetuned using clean speech signals of the test user. This method relies entirely on the finetuning process for personalization. On the other hand, the proposed methods provide various SSL options to pretrain the model using noisy, but speaker-specific speech, which serve a better initialization point for the subsequent finetuning process, leading to better SE performance. The pretrained models can also conduct a certain level of SE as a ZSL model, while the FSL-based finetuning tends to improve the pretrained model.

Our utilization of SSL stems from the assumption that *noisy* utterances from the target speaker $\tilde{\boldsymbol{s}} \in \tilde{\mathbb{S}}_{\mathbf{p}\text{-tr}}$ are much more available than clean ones, i.e., $|\tilde{\mathbb{S}}_{\mathbf{p}\text{-tr}}| \gg |\mathbb{S}_{\mathbf{f}\text{-tr}}|$. Our proposed pretraining methods aim to exploit these noisy observations as much as possible to learn the specificity of the test-time speaker. As is the case with SSL methods, the model parameters will be initialized via a pretext task, which is a made-up task that does not reflect a true speech enhancement function.

We assert, for example, that smart devices are likely to accrue many noisy recordings from the test-time speaker over time and with usage, i.e., $|\tilde{\mathbb{S}}_{\mathbf{p}\text{-tr}}| \gg |\mathbb{S}_{\mathbf{f}\text{-tr}}|$. Although we want to exploit these in-the-wild recordings $|\tilde{\mathbb{S}}_{\mathbf{p}\text{-tr}}|$, we do not know whether the observations are clean or noisy, i.e., the data is unlabeled. Therefore, we have to assume that $|\tilde{\mathbb{S}}_{\mathbf{p}\text{-tr}}|$ holds contaminated versions of some unobserved target clean speech signal $|\mathbb{S}_{\mathbf{p}\text{-tr}}|$. We refer to this unobserved contamination process as *premixture*. If we consider a hypothetical set of premixture noises $\boldsymbol{m} \in \mathbb{M}_{\text{tr}}$, then we can form a basic framework for premixture, i.e., $\tilde{\boldsymbol{s}} = \boldsymbol{s} + \boldsymbol{m}$. Because the true speech and noise signals which compose $\tilde{\boldsymbol{s}}$ are unknown, the premixture observations are unsuitable for conventional fully-supervised speech enhancement tasks nor for finetuning-based personalization.

Fig. 3 summarizes the training procedure of the baseline generalist-based pretraining, comparing it to our proposed SSL-based pretraining. Both approaches to personalization are based on transfer learning. Finetuning via FSL improves the baseline SE performance, exposing the generalist to the target speaker. However, the proposed SSL methods already achieve a certain level of personalization by using noisy speech signals of the target speaker, leading to a better ZSL solution than the generalist.

A. Training a Specialist through Pseudo Speech Enhancement

Depending on the user's test-time acoustic conditions, it is likely that the premixture noise component $\boldsymbol{m}$ has a loudness that varies over time. Then it follows that, at certain times, this premixture noise may be quiet enough such that the test-time speaker's voice $\boldsymbol{s}$ is the dominant signal. In these cases where there is a favorable premixture with a high signal-to-noise ratio (SNR), the noisy speech utterances $\tilde{\boldsymbol{s}}$ could be used as *pseudo* speech references. We can then formulate a pretraining process which we call *pseudo speech enhancement* (PseudoSE). This method operates using "doubly-degraded" artificial mixture signals. We construct the model inputs by sampling the abundant premixture set $\tilde{\boldsymbol{s}} \in \tilde{\mathbb{S}}_{\mathbf{p}\text{-tr}}$ and injecting the additional training noises $\boldsymbol{n} \in \mathbb{N}_{\text{tr}}$, i.e., $\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{s}} + \boldsymbol{n}$. This is a double-degradation process as $\tilde{\boldsymbol{s}}$ has been already contaminated by $\tilde{\boldsymbol{m}}$.

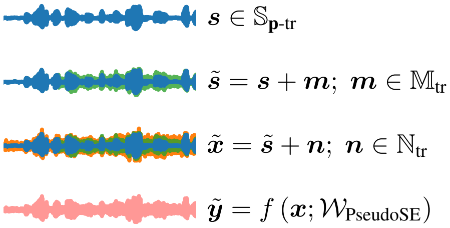

$\boldsymbol{s} \in \mathbb{S}_{\mathbf{p}\text{-tr}}$
Premixture
$\tilde{\boldsymbol{s}} = \boldsymbol{s} + \boldsymbol{m};\ \boldsymbol{m} \in \mathbb{M}_{\text{tr}}$
Noise Injection
$\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{s}} + \boldsymbol{n};\ \boldsymbol{n} \in \mathbb{N}_{\text{tr}}$
Enhancement
$\tilde{\boldsymbol{y}} = f\left(\boldsymbol{x}; \mathcal{W}_{\text{PseudoSE}}\right)$

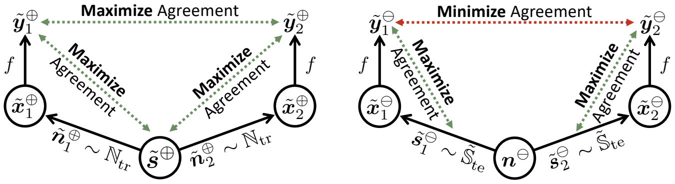
Fig. 4. Single-speaker (self-supervised) pseudo speech enhancement setup. The training target is pseudo-clean speech $\tilde{\boldsymbol{s}}$, therefore the model parameters $\mathcal{W}_{\text{PseudoSE}}$ are iteratively updated to minimize the loss function $\mathcal{L}_{\text{PseudoSE}}$. We simulate the process of sampling from the in-the-wild recordings, $\tilde{\boldsymbol{s}} \in \tilde{\mathbb{S}}_{\mathbf{p}\text{-tr}}$, using the premixture data transformation.

Consequently, the self-supervised model is a mapping function f with parameters $\mathcal{W}_{\text{PseudoSE}}$ that is trained to remove the injection noise and recover the pseudo speech target, i.e., $f(\tilde{\boldsymbol{x}}; \mathcal{W}_{\text{PseudoSE}}) = \tilde{\boldsymbol{y}} \approx \tilde{\boldsymbol{s}}$. Note that this self-supervised objective is not equivalent to the fully-supervised objective due to the difference in training target. f is only trained to recover the premixture utterance $\tilde{\boldsymbol{s}}$, therefore it is not a true speech enhancement function, i.e., $\mathcal{W}_{\text{PseudoSE}} \neq \mathcal{W}_{\text{SE}}$.

$$\mathcal{L}_{\text{PseudoSE}} = \mathcal{E}(\tilde{\boldsymbol{y}} \parallel \tilde{\boldsymbol{s}}) \tag{4}$$

$$\mathcal{W}_{\text{PseudoSE}} \leftarrow \underset{\mathcal{W}_{\text{PseudoSE}}}{\arg\min}\, \mathcal{L}_{\text{PseudoSE}} \tag{5}$$

Fig. 4 shows a visualization of the PseudoSE pretraining process. After the model parameters $\mathcal{W}_{\text{PseudoSE}}$ are learned, we may apply finetuning using known clean speech from the scarce set $\mathbb{S}_{\mathbf{f}\text{-tr}}$. In this FSL personalization context, the training targets are genuine clean speech utterances $\boldsymbol{s} \in \mathbb{S}_{\mathbf{f}\text{-tr}}$. Therefore, the parameters from the pseudo enhancement function $\mathcal{W}_{\text{PseudoSE}}$ are iteratively updated in order to fit a real speech enhancement


Maximize Agreement
$\tilde{\boldsymbol{y}}_1^\oplus$
$\tilde{\boldsymbol{y}}_2^\oplus$
Maximize Agreement
Maximize Agreement
f
f
$\tilde{\boldsymbol{x}}_1^\oplus$
$\tilde{\boldsymbol{x}}_2^\oplus$
$\tilde{\boldsymbol{n}}_1^\oplus \sim \mathbb{N}_{\text{tr}}$
$\tilde{\boldsymbol{s}}^\oplus$
$\tilde{\boldsymbol{n}}_2^\oplus \sim \mathbb{N}_{\text{tr}}$
Minimize Agreement
$\tilde{\boldsymbol{y}}_1^\ominus$
$\tilde{\boldsymbol{y}}_2^\ominus$
Maximize Agreement
Maximize Agreement
f
f
$\tilde{\boldsymbol{x}}_1^\ominus$
$\tilde{\boldsymbol{x}}_2^\ominus$
$\tilde{\boldsymbol{s}}_1^\ominus \sim \tilde{\mathbb{S}}_{\text{te}}$
$\boldsymbol{n}^\ominus$
$\tilde{\boldsymbol{s}}_2^\ominus \sim \tilde{\mathbb{S}}_{\text{te}}$

(a) Positive Pairs

(b) Negative Pairs

Fig. 5. The proposed framework for contrastive mixtures. Solid lines indicate signal path while dashed lines show loss terms.

function. Once again, the finetuning loss function is equivalent to Eq. (2) using the speaker-specific mixtures.

There are trade-offs to note when using self-supervised learning. On one hand, the success of PseudoSE pretraining is bounded by the noisiness of $\tilde{s}$, the impure training targets. But on the other hand, this pretraining scheme uses data derived only from the target speaker, thereby bypassing the need for generalization. Unlike the baseline method, which recasts a generalist as a specialist, PseudoSE pretraining directly develops a specialist model. However, the PseudoSE model could under perform when compared to a hypothetical fully-supervised model exposed to ample clean speech from the target speaker. If finetuning is not possible, the PseudoSE model could serve as a zero-shot solution on its own. But if finetuning is possible, we claim that PseudoSE serves as a more optimal pretraining scheme as opposed to the baseline speaker-agnostic SE.

B. Training a Specialist through Contrastive Mixtures

We hypothesize that the quality of the pretraining procedure greatly impacts how the downstream denoising model will personalize. Even if the premixed noisy speech set $\tilde{\mathbb{S}}_{\mathbf{p}\text{-tr}}$ and the deformation noise set $\mathbb{N}_{\text{tr}}$ are large, the quality of the features learned through PseudoSE are bounded by how noisy $\tilde{\mathbb{S}}_{\mathbf{p}\text{-tr}}$ really is. Our proposed *contrastive mixtures* (CM) pretraining procedure addresses this by employing a pairwise contrastive learning mechanism. In the CM framework, the denoising model $f(\cdot)$ pretrains over *pairs* of mixtures ($\tilde{\boldsymbol{x}}_1$, $\tilde{\boldsymbol{x}}_2$) and outputs pseudo-cleaned estimates ($\tilde{\boldsymbol{y}}_1$, $\tilde{\boldsymbol{y}}_2$). We create two kinds of mixture pairs, *positive* and *negative*, which are illustrated in Figure 5.

In a positive pair, both input examples ($\tilde{\boldsymbol{x}}_1^\oplus$, $\tilde{\boldsymbol{x}}_2^\oplus$) share the same premixture source $\tilde{\boldsymbol{s}}^\oplus$, but are differently deformed; that is, the mixing process makes the input pair dissimilar. Therefore, in addition to maximizing the similarities between estimates and source ($\tilde{\boldsymbol{y}}_1^\oplus$ to $\tilde{\boldsymbol{s}}^\oplus$ and $\tilde{\boldsymbol{y}}_2^\oplus$ to $\tilde{\boldsymbol{s}}^\oplus$), the model $f_{\text{CM}}(\cdot)$ must also satisfy the contrastive objective based on the fact that $\tilde{\boldsymbol{y}}_1^\oplus$ and $\tilde{\boldsymbol{y}}_2^\oplus$ stemmed from the same pseudo source. We express these objectives as a positive pair loss function $\mathcal{L}_p$ in the following form:

$$\mathcal{L}_p = \mathcal{E}(\tilde{\boldsymbol{s}}^\oplus||\tilde{\boldsymbol{y}}_1^\oplus) + \mathcal{E}(\tilde{\boldsymbol{s}}^\oplus||\tilde{\boldsymbol{y}}_2^\oplus) + \lambda_p\big[\mathcal{E}(\tilde{\boldsymbol{y}}_1^\oplus||\tilde{\boldsymbol{y}}_2^\oplus)\big], \tag{6}$$

where λ_p scales the contribution of the contrastive loss term.

In a negative pair, each mixture is made from a *different* pseudo source ($\tilde{\boldsymbol{s}}_1^\ominus \neq \tilde{\boldsymbol{s}}_2^\ominus$), but with a shared deformation, i.e., $\tilde{\boldsymbol{x}}_1^\ominus = \tilde{\boldsymbol{s}}_1^\ominus + \boldsymbol{n}^\ominus$ and $\tilde{\boldsymbol{x}}_2^\ominus = \tilde{\boldsymbol{s}}_2^\ominus + \boldsymbol{n}^\ominus$; in other words, the negative pair mixing process makes the originally different inputs more similar to one another. Accordingly, in addition to the source-wise denoising objectives, the dissimilarity between the estimates $\tilde{\boldsymbol{y}}_1^\ominus$ and $\tilde{\boldsymbol{y}}_2^\ominus$ must be taken into consideration. We express these objectives as a negative pair loss function $\mathcal{L}_n$ in the following form:

$$\begin{aligned}\mathcal{L}_n = {} & \mathcal{E}(\tilde{\boldsymbol{s}}_1^\ominus||\tilde{\boldsymbol{y}}_1^\ominus) + \mathcal{E}(\tilde{\boldsymbol{s}}_2^\ominus||\tilde{\boldsymbol{y}}_2^\ominus) \\ & + \lambda_n\big[\max\big(\mathcal{E}(\tilde{\boldsymbol{s}}_1^\ominus||\tilde{\boldsymbol{s}}_2^\ominus), \mathcal{E}(\tilde{\boldsymbol{y}}_1^\ominus||\tilde{\boldsymbol{y}}_2^\ominus)\big)\big],\end{aligned} \tag{7}$$

where λ_n controls the contribution of the contrastive loss term. Note that the $\max$ function sets up the bound for the disagreement term $\mathcal{E}(\tilde{\boldsymbol{y}}_1^\ominus||\tilde{\boldsymbol{y}}_2^\ominus)$ comparing it with the “desired” disagreement level of the target pseudo sources $\mathcal{E}(\tilde{\boldsymbol{s}}_1^\ominus||\tilde{\boldsymbol{s}}_2^\ominus)$, rather than enforcing an unbounded disagreement.

Both $\mathcal{L}_p$ and $\mathcal{L}_n$ consist of two terms: the source-to-estimate errors and the estimate-to-estimate errors. The former term characterizes the main speech enhancement loss, while the latter term provides the proposed contrastive regularization. The model ultimately minimizes the sum of these two losses,

$$\mathcal{L}_{\text{CM}} = \sum_{t=1}^{T}\mathcal{L}_p(t) + \sum_{t=1}^{T}\mathcal{L}_n(t) \tag{8}$$

$$\mathcal{W}_{\text{CM}} \leftarrow \operatorname*{arg\,min}_{\mathcal{W}_{\text{CM}}} \mathcal{L}_{\text{CM}}, \tag{9}$$

where T is the number of positive or negative pairs within the batch and $\mathcal{L}_p(t)$ and $\mathcal{L}_n(t)$ denote the loss for the t-th pair. If the regularizing contrastive terms are omitted, i.e., by setting $\lambda_p = 0$ and $\lambda_n = 0$, it can be shown that $\mathcal{L}_{\text{CM}}$ reduces to Eq. (4). Four our experiments, we set T to be half of the batch size. To find optimal choices for λ_p and λ_n, we run an ablation study as described in Sec. VI-A.

Our proposed CM approach differs from the SimCLR model [19] in multiple regards: (a) it uses a more sophisticated noise injection for data augmentation to mimic the real-world noisy speech mixture generation process, i.e. by using non-stationary noise sources; (b) the introduction of the negative pairs more precisely reflects the source separation concept underlying our SE problem and yields a more discriminative feature than a positive pair only; and, (c) having the traditional SE loss term prevents trivial solutions to the contrastive loss-only case—estimating very similar $\tilde{\boldsymbol{y}}_1^\ominus$ and $\tilde{\boldsymbol{y}}_2^\ominus$ that do not recover the pseudo sources.

C. Data Purification

When it comes to fully-supervised pretraining, we know that the target signals are clean because they originate from the large labeled dataset $\mathbb{G}_{\text{tr}}$. However, the target signals' cleanliness is ambiguous in the case of self-supervised pretraining, which utilizes $\tilde{\mathbb{S}}_{\mathbf{p}\text{-tr}}$ as the pseudo source. Based on our formulation of the premixture process in Fig. 4, two factors determine whether the pseudo sources $\tilde{s}$ are too degraded to be usable. These are: the sparsity of premixture noise $\boldsymbol{m}$, as well as the segmental SNR between $\boldsymbol{s}$ and $\boldsymbol{m}$. For example, if $\boldsymbol{m}$ is sufficiently sparse, portions of $\tilde{s}$ may contain near-clean speech. Considering all the available noisy utterances $\tilde{s} \in \tilde{\mathbb{S}}_{\mathbf{p}\text{-tr}}$, we hypothesize that utterances with a higher SNR may serve as more useful target signals than other noisier utterances, even if none of them

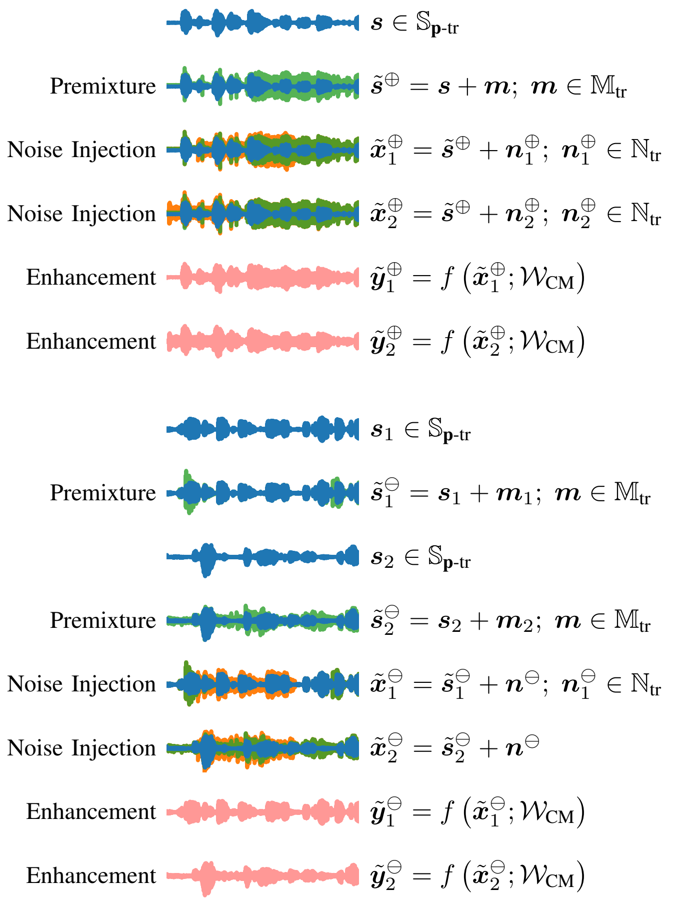

$\boldsymbol{s} \in \mathbb{S}_{\textbf{p-tr}}$
Premixture $\tilde{\boldsymbol{s}}^{\oplus} = \boldsymbol{s} + \boldsymbol{m};\ \boldsymbol{m} \in \mathbb{M}_{\text{tr}}$
Noise Injection $\tilde{\boldsymbol{x}}_1^{\oplus} = \tilde{\boldsymbol{s}}^{\oplus} + \boldsymbol{n}_1^{\oplus};\ \boldsymbol{n}_1^{\oplus} \in \mathbb{N}_{\text{tr}}$
Noise Injection $\tilde{\boldsymbol{x}}_2^{\oplus} = \tilde{\boldsymbol{s}}^{\oplus} + \boldsymbol{n}_2^{\oplus};\ \boldsymbol{n}_2^{\oplus} \in \mathbb{N}_{\text{tr}}$
Enhancement $\tilde{\boldsymbol{y}}_1^{\oplus} = f\left(\tilde{\boldsymbol{x}}_1^{\oplus}; \mathcal{W}_{\text{CM}}\right)$
Enhancement $\tilde{\boldsymbol{y}}_2^{\oplus} = f\left(\tilde{\boldsymbol{x}}_2^{\oplus}; \mathcal{W}_{\text{CM}}\right)$
$\boldsymbol{s}_1 \in \mathbb{S}_{\textbf{p-tr}}$
Premixture $\tilde{\boldsymbol{s}}_1^{\ominus} = \boldsymbol{s}_1 + \boldsymbol{m}_1;\ \boldsymbol{m} \in \mathbb{M}_{\text{tr}}$
$\boldsymbol{s}_2 \in \mathbb{S}_{\textbf{p-tr}}$
Premixture $\tilde{\boldsymbol{s}}_2^{\ominus} = \boldsymbol{s}_2 + \boldsymbol{m}_2;\ \boldsymbol{m} \in \mathbb{M}_{\text{tr}}$
Noise Injection $\tilde{\boldsymbol{x}}_1^{\ominus} = \tilde{\boldsymbol{s}}_1^{\ominus} + \boldsymbol{n}^{\ominus};\ \boldsymbol{n}_1^{\ominus} \in \mathbb{N}_{\text{tr}}$
Noise Injection $\tilde{\boldsymbol{x}}_2^{\ominus} = \tilde{\boldsymbol{s}}_2^{\ominus} + \boldsymbol{n}^{\ominus}$
Enhancement $\tilde{\boldsymbol{y}}_1^{\ominus} = f\left(\tilde{\boldsymbol{x}}_1^{\ominus}; \mathcal{W}_{\text{CM}}\right)$
Enhancement $\tilde{\boldsymbol{y}}_2^{\ominus} = f\left(\tilde{\boldsymbol{x}}_2^{\ominus}; \mathcal{W}_{\text{CM}}\right)$

Fig. 6. Single-speaker (self-supervised) contrastive mixtures setup. With positive pairs, there is a single training target, pseudo source $\tilde{\boldsymbol{s}}^{\oplus}$. With negative pairs, there are two different training targets, pseudo sources $\tilde{\boldsymbol{s}}_1^{\ominus}$ and $\tilde{\boldsymbol{s}}_2^{\ominus}$. The model parameters $\mathcal{W}_{\text{CM}}$ are iteratively updated to minimize the loss function $\mathcal{L}_{\text{CM}}$.

are completely clean. The proposed self-supervised pretraining methods can benefit from knowing where the cleaner frames within $\tilde{\boldsymbol{s}}$ may be.

For that reason, we put forward a *data purification* (DP) pipeline. In essence, we modify the discrepancy function $\mathcal{E}$ to incorporate a weighting vector $\boldsymbol{p}$. To generate this DP weighting vector, we first train a separate neural network that estimates the frame-by-frame SNR of the premixtures. The quality estimator network h is a regressive model trained over a diverse set of training speakers and noises (i.e., $\mathbb{G}_{\text{tr}}$ and $\mathbb{N}_{\text{tr}}$). It outputs a vector of segmental SNRs, $\hat{\boldsymbol{\alpha}}$. Hence, the network h works as a general-purpose speech quality estimator, that has no prior knowledge of the test-time speaker or the test-time noisy environment. Given an estimate signal $\hat{\boldsymbol{v}}$ and a target signal $\boldsymbol{v}$ both of length L, their residual is $\boldsymbol{r} = \boldsymbol{v} - \hat{\boldsymbol{v}}$, and the frame-by-frame/segmental SNR (SegSNR) is defined as:

$$\text{SegSNR}_j(\boldsymbol{v}, \hat{\boldsymbol{v}}) = 10\log_{10}\left[\frac{\sum_{i=Hj}^{Hj+N-1}\left(w_i - Hjv_i\right)^2}{\sum_{i=Hj}^{Hj+N-1}\left(w_i - Hjr_i\right)^2}\right], \tag{10}$$

where N is the frame size, H is the hop size, j is a zero-based frame index (i.e. $0 \leq j \leq \lceil \frac{L}{H} \rceil - 1$), and vector $\boldsymbol{w}$ comes from the Hann window function of length N. We then formulate the training process of the SNR Predictor network as follows:

$$\begin{aligned} \boldsymbol{x} &= \boldsymbol{s} + \boldsymbol{n}; \quad \boldsymbol{s} \in \mathbb{G}_{\text{tr}},\ \boldsymbol{n} \in \mathbb{N}_{\text{tr}} \\ \boldsymbol{\alpha} &= \text{SegSNR}(\boldsymbol{s}, \boldsymbol{x}) \\ \hat{\boldsymbol{\alpha}} &= h(\boldsymbol{x}; \mathcal{W}_h) \\ \mathcal{W}_h &\leftarrow \underset{\mathcal{W}_h}{\arg\min}\ \text{MSE}(\hat{\boldsymbol{\alpha}}, \boldsymbol{\alpha}), \end{aligned} \tag{11}$$

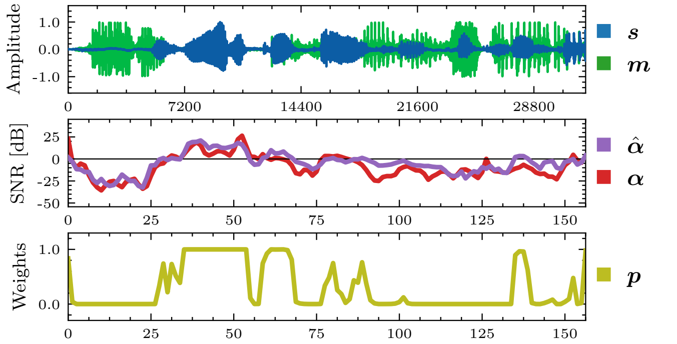


Fig. 7. Illustration of the SNR predictor inputs and outputs. The first subplot features an example premixture/pseudo source $\tilde{\boldsymbol{s}}$. In the second subplot, the SNR predictor network h estimates the frame-wise (i.e., segmental) SNR of the premixture. The training objective of h is to minimize the loss between estimates $\hat{\boldsymbol{\alpha}}$ and targets $\boldsymbol{\alpha}$. The third subplot shows the frame-by-frame SNR estimates converted into weights using the logistic function, i.e. $\boldsymbol{p} = \sigma(h(\tilde{\boldsymbol{s}}))$.

Note that the SNR predictor inputs are of length L, but its outputs are of length $\lceil \frac{L}{H} \rceil$; in other words, $\boldsymbol{x}$'s length is measured in samples but $\hat{\boldsymbol{\alpha}}$'s length is measured in frames.

We can now apply a DP step to improve the reliability of the pseudo-target $\tilde{\boldsymbol{s}}$ during PseudoSE and CM pretraining. With each iteration of pretraining, the SNR predictor h first analyzes the input premixtures to estimate frame-wise SNRs, $\hat{\boldsymbol{\alpha}} = h(\tilde{\boldsymbol{s}})$. Next, we apply the logistic function σ to the $\hat{\boldsymbol{\alpha}}$ logits in order to obtain frame-by-frame weights:

$$\boldsymbol{p} = \sigma(\hat{\boldsymbol{\alpha}}) = \frac{1}{1 + e^{-\hat{\boldsymbol{\alpha}}}}. \tag{12}$$

Lastly, we modify both PseudoSE and CM pretraining procedures to use $\mathcal{E}_{\text{DP}}$ which promotes speech-prominent frames in the loss function. To that end, we re-write Eq. (10) to incorporate the frame-by-frame weights $\boldsymbol{p}$. That is, the signal discrepancy is computed between windowed segments, which are then weighted by $\boldsymbol{p}$ and finally averaged across all frames. Because this is a neural network loss function to be minimized, we use the negative of weighted segmental SNR, which we denote as $\overline{\text{SegSNR}}$.

$$\begin{aligned} \mathcal{E}_{\text{DP}}(\tilde{\boldsymbol{y}} \parallel \tilde{\boldsymbol{s}}) &= \overline{\text{SegSNR}}(\tilde{\boldsymbol{y}}, \tilde{\boldsymbol{s}}; \boldsymbol{p}) \\ &= -\frac{1}{J}\sum_{j=0}^{J-1} p_j \left[10\log_{10}\frac{\sum_{i=Hj}^{Hj+N-1}\left(w_{i-Hj}\tilde{s}_i\right)^2}{\sum_{i=Hj}^{Hj+N-1}\left(w_{i-Hj}\tilde{r}_i\right)^2}\right] \end{aligned} \tag{13}$$

Here, J is the number of frames $\lceil \frac{L}{H} \rceil$. Additionally, the residual vector is defined as $\tilde{\boldsymbol{r}} = \tilde{\boldsymbol{s}} - \tilde{\boldsymbol{y}}$. This regressive model h does not need to have pinpoint accuracy; as shown in Fig. 7, as long as $\hat{\boldsymbol{\alpha}}$ decently approximates $\boldsymbol{\alpha}$, the weights $\boldsymbol{p}$ will accurately reflect the position of speech-prominent frames in the data.

If we substitute $\mathcal{E}_{\text{DP}}$ for $\mathcal{E}$ into the original PseudoSE loss function—Eq. 4—we obtain a new data purified loss function:

$$\mathcal{L}_{\text{PseudoSE+DP}} = \mathcal{E}_{\text{DP}}(\tilde{\boldsymbol{y}} \parallel \tilde{\boldsymbol{s}}). \tag{14}$$

Note that the slope of the logistic function could be further controlled by using an additional temperature weight applied to $\hat{\boldsymbol{\alpha}}$, which we opt not to investigate to focus more on the main contributions.

Though substituting $\mathcal{E}_{\text{DP}}$ within the PseudoSE loss function is straightforward, it requires more nuance with the CM loss function. CM utilizes pairwise inputs, so therefore, we must compute pairwise weights as well.

$$\boldsymbol{p}^{\oplus} = \sigma(h(\tilde{\boldsymbol{s}}^{\oplus})),\ \ \boldsymbol{p}_1^{\ominus} = \sigma(h(\tilde{\boldsymbol{s}}_1^{\ominus})),\ \ \boldsymbol{p}_2^{\ominus} = \sigma(h(\tilde{\boldsymbol{s}}_2^{\ominus})) \tag{15}$$

Specifically in the case of positive pairs, the underlying pseudo source is the same, which is why there is only a single set of weights $\boldsymbol{p}^{\oplus}$. Negative pairs are made up of two pseudo sources, so there are two sets of weights. For the negative pair estimate-to-estimate losses, we use the product of the two weight vectors, i.e. $\boldsymbol{p}^{\ominus} = \boldsymbol{p}_1^{\ominus} \cdot \boldsymbol{p}_2^{\ominus}$. Using the appropriate weights for every term, we rewrite Eq. (6) and Eq. (7) as:

$$\begin{aligned}\mathcal{L}_{p\text{+DP}} = & \overline{\text{SegSNR}}(\tilde{\boldsymbol{y}}_1^{\oplus}, \tilde{\boldsymbol{s}^{\oplus}}; \boldsymbol{p}^{\oplus}) + \\ & \overline{\text{SegSNR}}(\tilde{\boldsymbol{y}}_2^{\oplus}, \tilde{\boldsymbol{s}^{\oplus}}; \boldsymbol{p}^{\oplus}) + \\ & \lambda_p \left[\overline{\text{SegSNR}}(\tilde{\boldsymbol{y}}_1^{\oplus}, \tilde{\boldsymbol{y}}_2^{\oplus}; \boldsymbol{p}^{\oplus}) \right]\end{aligned} \tag{16}$$

$$\begin{aligned}\mathcal{L}_{n\text{+DP}} = & \overline{\text{SegSNR}}(\tilde{\boldsymbol{y}}_1^{\ominus}, \tilde{\boldsymbol{s}_1^{\ominus}}; \boldsymbol{p}_1^{\ominus}) + \\ & \overline{\text{SegSNR}}(\tilde{\boldsymbol{y}}_2^{\ominus}, \tilde{\boldsymbol{s}_2^{\ominus}}; \boldsymbol{p}_2^{\ominus}) + \\ & \lambda_n \big[\max \big(\overline{\text{SegSNR}}(\tilde{\boldsymbol{s}}_1^{\ominus}, \tilde{\boldsymbol{s}}_2^{\ominus}; \boldsymbol{p}^{\ominus}), \\ & \qquad \overline{\text{SegSNR}}(\tilde{\boldsymbol{y}}_1^{\ominus}, \tilde{\boldsymbol{y}}_2^{\ominus}; \boldsymbol{p}^{\ominus}) \big) \big]\end{aligned} \tag{17}$$

The data-purified positive and negative loss functions may now be substituted in Eq. (8) to obtain the overall CM+DP loss function:

$$\mathcal{L}_{\text{CM+DP}} = \sum_{t=1}^{T} \mathcal{L}_{p\text{+DP}}(t) + \sum_{t=1}^{T} \mathcal{L}_{n\text{+DP}}(t). \tag{18}$$

V. Experiment Setup

In our experiments, we compare the baseline fully-supervised approach with the two proposed self-supervised approaches for training a personalized speech enhancement model. Note that there are two rounds of model training (Fig. 3): one round that pretrains the model, and another "finetuning" round that only uses the available clean target speaker data (either 5 sec or 30 sec). We also assess the benefits of adding the data purification step to both self-supervised methods. We use the following shorthand notation to refer to each pretraining method:

- **SE**: Models trained to minimize Eq. (2). This is our generalist baseline, the speaker-agnostic speech enhancement system. It generalizes well only if its model capacity is large enough.
- **PseudoSE**: Models trained to minimize Eq. (4). The proposed self-supervised method relies solely on noisy speaker-specific data $\tilde{\mathbb{S}}_{\mathbf{p}\text{-tr}}$.
- **PseudoSE+DP**: Models trained to minimize Eq. (14). This method refines the prior method through data purification. That is, the model uses a weighted segmental MSE as its discrepancy function in order to filter out noise-dominant frames within $\tilde{\mathbb{S}}_{\mathbf{p}\text{-tr}}$.
- **CM**: Models trained to minimize Eq. (8). This self-supervised method uses pairwise inputs that share either the same pseudo source or injection noise. CM provides additional regularization to PseudoSE through the contrastive loss terms.
- **CM+DP**: Models trained to minimize Eq. (18). The pairwise weights inform the model of the mutual speech-dominant frames, thereby focusing the contrastive regularization specifically wherever the test-time speech is prominent.

A. Datasets

Table I provides a glossary of all the datasets and their notation used throughout this paper. Note that we subscript all datasets with either 'tr', 'vl', or 'te' to indicate training, validation, or test partitions respectively. For this paper, we limit the scope of personalization specifically regarding the test-time speaker and not the test-time environment. The extension of our methods towards environment adaptation is straightforward.

In order to report objective signal improvement results, we designed experiments that simulate the personalization context. We therefore artificially mix signals from three publicly-available audio datasets: we use LibriSpeech [48] for clean speech recordings, FSD50K [49] for premixture noises, and MUSAN [50] for additionally injected noises.

Out of the LibriSpeech *train-clean-100* subset, we set aside 20 speakers to be the personalization targets; in other words, there are $K = 20$ speaker-specific datasets $\mathbb{S}^{(k)}$ where $k \in \{1, \ldots, K\}$. We omit the speaker index k going forward to simplify notation. The remaining speakers within Librispeech's *train-clean-100* and *train-clean-360* subsets are consolidated into the speaker-agnostic dataset $\mathbb{G}$. For all speech and noise corpora, we discard audio files shorter than 4 sec and resample everything to 16 kHz.

We partition each speaker-specific dataset $\mathbb{S}$ into five sets as shown in Table I. The utterances are sorted by duration and grouped such that approximately 30 sec are available for testing the model ($\mathbb{S}_{\text{te}}$), 30 sec for validating finetuned models ($\mathbb{S}_{\mathbf{f}\text{-vl}}$), 60 sec for FSL-based finetuning ($\mathbb{S}_{\mathbf{f}\text{-tr}}$), and 30 sec to validate the self-supervised pretraining methods ($\mathbb{S}_{\mathbf{p}\text{-vl}}$). The remaining 22.5 min are used for pretraining ($\mathbb{S}_{\mathbf{p}\text{-tr}}$). Subsequently, for each of the 20 personalization targets, a test set is constructed using 100 mixtures that combine $\mathbb{S}_{\text{te}}$ with $\mathbb{N}_{\text{te}}$.

$\mathbb{M}_{\text{tr}}$ and $\mathbb{M}_{\text{vl}}$ follow the train and val splits provided in FSD50K's *dev* folder. Using the FSD50K provided tags, we omit files tagged as either "speech" or "music".

The unseen test-time noises, $\mathbb{N}_{\text{te}}$, are derived from MUSAN's *sound-bible* folder. Using MUSAN's *free-sound* folder, sixty random noises are set aside for $\mathbb{N}_{\text{vl}}$ and the remaining noises make up $\mathbb{N}_{\text{tr}}$.

These datasets are carefully chosen and arranged to represent our use-case scenarios. First, we need a large dataset $\mathbb{G}$ to

TABLE I
GLOSSARY OF DATASETS PAIRED WITH EXPERIMENT-SPECIFIC CORPORA.

Set	Subset	Duration	Quantity	Description	Corpus
$\mathbb{G}$	$\mathbb{G}_{\text{tr}}$ $\mathbb{G}_{\text{vl}}$	443 h 8 h	1,132 spkrs 20 spkrs	Clean speech from many anonymous speakers	LibriSpeech [48]
$\mathbb{S}$	$\mathbb{S}_{\mathbf{p}\text{-tr}}$ $\mathbb{S}_{\mathbf{p}\text{-vl}}$	22.5 min /spkr 30 sec /spkr	20 spkrs	Target speaker's noisy speech (corrupted by $\mathbb{M}$), referred to as the set of *premixture* data	LibriSpeech [48]
	$\mathbb{S}_{\mathbf{f}\text{-tr}}$ $\mathbb{S}_{\mathbf{f}\text{-vl}}$	60 sec /spkr 30 sec /spkr		Target speaker's provided clean speech (only available in FSL context)	
	$\mathbb{S}_{\text{te}}$	30 sec /spkr		Set-aside clean speech from the target speaker used only for objective model evaluation	
$\mathbb{M}$	$\mathbb{M}_{\text{tr}}$ $\mathbb{M}_{\text{vl}}$	48 h 7 h	13,339 noises 1,929 noises	Premixture noises corrupting the majority of target speaker's utterances; on their own, inaccessible during model training	FSD50K [49]
$\mathbb{N}$	$\mathbb{N}_{\text{tr}}$ $\mathbb{N}_{\text{vl}}$	5 h 0.5 h	616 noises 60 noises	Injection noises used during model pretraining and fine-tuning	MUSAN [50]
	$\mathbb{N}_{\text{te}}$	0.5 h	60 noises	Injection noises never seen during any model training, used to prepare target speaker-specific test sets	

encompass diverse speaker characteristics. Second, we ensure that the 20 personalization target speakers have enough clean speech signals $\mathbb{S}_{\mathbf{p}\text{-tr}}$ in order to simulate the abundant premixture signals $\tilde{\mathbb{S}}_{\mathbf{p}\text{-tr}}$. The premixture noise sources $\mathbb{M}_{\text{tr}}$ are also very diverse so as to simulate various acoustic environment the user can be situated in. Tallying the unique FSD50K audio tags, our experiment simulates each of the 20 target speakers being degraded by approximately 160 noise types. Through the premixture process, we combine $\boldsymbol{s}$ and $\boldsymbol{m}$ such that the SNR is uniformly random between 0 dB to 15 dB. Psychoacoustic research has shown that this SNR range describes many real-world sound environments [51], [52]. Lastly, mixtures, which are made using the injection noise set $\mathbb{N}$, have SNRs chosen uniformly at random between -5 dB to 5 dB.

There are other choices of speech datasets, besides Librispeech, which contain real-world recordings of in-the-wild noisy speech, e.g., AudioSet [53]. Although our proposed self-supervised training methods are intended for in-the-wild data, it is often the case that such datasets do not possess enough noisy recordings from a single consistent speaker. More importantly, in order for us to report objective signal improvement, we require ground-truth clean speech recordings from the test-time speaker. Therefore, our experiments simulate the personalization problem through the three separate corpora, constructing numerous artificial mixtures and premixtures.

B. Metrics

With our experiments, we report three metrics frequently used in speech enhancement research: SDR [46], PESQ [54], and extended STOI [55]. Unlike the objective measurement SDR, the latter two are perceptual metrics that highly correlate to speech intelligibility. As all of our loss functions are SDR-based, our models in this experiment do not explicitly optimize for intelligibility. Each one of the 20 target speakers has their own test set, made up of 100 mixtures with input SNR between -5 dB to 5 dB. All three metrics are computed between the estimate signals and their corresponding target signals.

TABLE II
LIST OF MODEL ARCHITECTURES, CONFIGURATIONS, AND SIZES.

Architecture	Size	Configuration	Params	MACs
Conv-TasNet	Large	$B_c = 64,\ H_c = 256$	1.0 M	8.4 G
	Medium	$B_c = 32,\ H_c = 128$	437.8 k	3.5 G
	Small	$B_c = 16,\ H_c = 64$	224.1 k	1.8 G
	Tiny	$B_c = 8,\ H_c = 32$	138.8 k	1.1 G

C. Neural Network Architectures

Well-established neural network approaches for speech enhancement utilize time-frequency masking. In order to overcome latency and phase reconstruction limitations, more recent neural network algorithms operate in an end-to-end manner, i.e., by learning a mapping directly between the time-domain input and output signals [56]–[58]. To that end, we assess the performance of generalist and specialist speech enhancement models using ConvTasNet (CTN), which is a popular fully-convolutional time-domain model for audio separation [10]. It operates as follows: first, the encoder module maps input waveforms into latent representations. Then, the separation module calculates a multiplicative mask that separates the target source. Lastly, the decoder module maps the masked latent features back to the time-domain, yielding estimate waveforms. The CTN architecture may be generalized to separate multiple audio sources; however, our separation module estimates only one mask to specifically separate speech from noise. With each size variant, we adjust the number of channels in the separation module's bottleneck (B_c) as well as the number of channels in convolutional blocks (H_c) such that the expansion ratio $H_c/B_c \approx 4$ [59].

As shown in Table II, we designed a tiny, small, medium, and large-sized variant of CTN such that the total number of trainable parameters is less than or equal to one million. MACs indicate the number of multiply-accumulate operations, correlating to computational complexity. As shown in prior work, personalized speech enhancement is a subset of the broader universal speech enhancement problem, therefore specialist models can achieve comparable performance to generalist models using fewer parameters [22], [35]. Through our experiments, we report the performance of the different sized variants to observe whether this model compression trend applies to the modern fully-convolutional models.

D. Implementation Details

All models were implemented using PyTorch [60] and trained on NVIDIA Tesla V100 graphics cards. We used the ConvTasNet implementation found in the Asteroid package [61]. All experiments have a fixed batch size of 64. We utilize the Adam optimizer [62] with an initial learning rate of 1e−3. When finetuning over clean speech data ($\mathbb{S}_{\text{f-tr}}$), the learning rate is instead 1e−4. For every 1000 mixtures processed, we compute SDR improvement averaged over a fixed set of 100 validation mixtures; the trial is terminated if the mean validation SDR does not improve after 100 000 further mixtures.

Using the described early stopping scheme, we observed various trends with regards to the training time. On average, generalist models trained over 1.4 M mixtures for all four sizes, whereas specialist models trained over 851 k, 803 k, 637 k, and 593 k mixtures for the Tiny, Small, Medium, and Large model sizes respectively. When these models undergo finetuning using 5 sec of clean speech, the specialists converge after seeing 6.4 k, 6.0 k, 5.7 k, and 5.2 k mixtures for the Tiny, Small, Medium, and Large model sizes respectively.

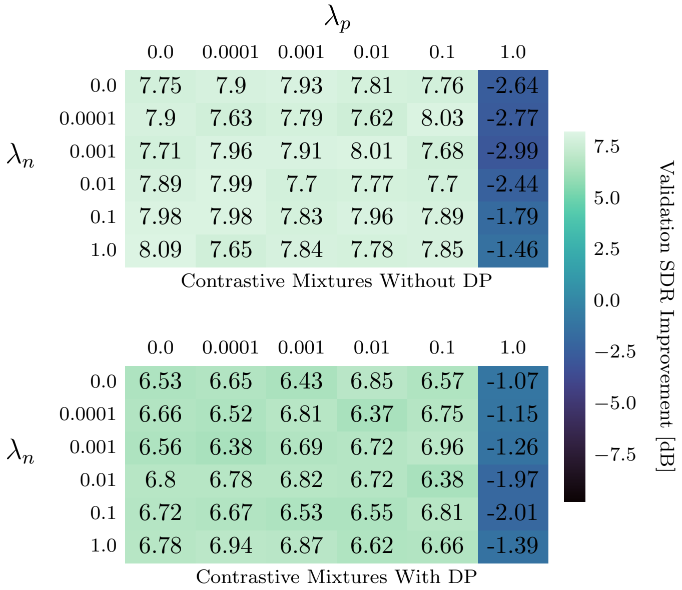
Source code for this experiment may be found at https://saige.sice.indiana.edu/research-projects/pse-ssl-dp.

VI. Experiment Results

A. Contrastive Mixtures Ablation Study

Prior to starting the full personalization experiment, we first determine optimal values for λ_p and λ_n which modulate the contrastive mixtures positive and negative loss terms—Eq. (6) and (7) with DP variants (16) and (17). Therefore, we run an ablation study of contrastive mixtures by performing a grid search over potential choices: 1, 1e−1, 1e−2, 1e−3, 1e−4, and 0. We can assess the effectiveness of the positive and negative pairs by setting either one of λ_n to λ_p to 0, respectively. For the purposes of the ablation study, we run experiments in which the personalized speech enhancement system is fixed as a small ConvTasNet as specified in Table II. This is done for three out of the twenty personalization target speakers from LibriSpeech. This results in 216 total trials, given that there are 36 λ combinations and 3 target speakers, plus the option for data purification to be enabled or disabled. We report the validation set signals' SDRs after pseudo-enhancement, averaged across the three speakers and across 100 validation premixtures utterances. In summary, a small ConvTasNet is trained over speaker-specific premixtures using a batch size of 64, a learning rate of 1e−3, and the **CM** loss function: either Eq. (8) or (18).

From Fig. 8, we observe that there are many working combinations of λ_p and λ_n, so long as $\lambda_p < 1$. This suggests that CM is robust to the hyperparameter selection. The top-left corner of both subplots represents models trained with the contrastive loss terms disabled—effectively, trained through PseudoSE. By scanning the left-most column and top-most row, we can see that the negative pair loss terms improve the model more significantly than the positive pair loss terms.



λ_n \ λ_p	0.0	0.0001	0.001	0.01	0.1	1.0
0.0	7.75	7.9	7.93	7.81	7.76	-2.64
0.0001	7.9	7.63	7.79	7.62	8.03	-2.77
0.001	7.71	7.96	7.91	8.01	7.68	-2.99
0.01	7.89	7.99	7.7	7.77	7.7	-2.44
0.1	7.98	7.98	7.83	7.96	7.89	-1.79
1.0	8.09	7.65	7.84	7.78	7.85	-1.46

λ_n \ λ_p	0.0	0.0001	0.001	0.01	0.1	1.0
0.0	6.53	6.65	6.43	6.85	6.57	-1.07
0.0001	6.66	6.52	6.81	6.37	6.75	-1.15
0.001	6.56	6.38	6.69	6.72	6.96	-1.26
0.01	6.8	6.78	6.82	6.72	6.38	-1.97
0.1	6.72	6.67	6.53	6.55	6.81	-2.01
1.0	6.78	6.94	6.87	6.62	6.66	-1.39

Fig. 8. Ablation study of the contrastive mixtures (CM) loss function, where we vary λ_p and λ_n to adjust the contribution of the positive and negative pair loss terms. Pseudo-enhancement is performed using the small ConvTasNet architecture, and results are averaged across three test-time speakers.

When pretraining without data purification, the most-optimal configuration happens to be with $\lambda_n = 1$ and $\lambda_p = 0$, yielding a 0.34 dB (or 4.4 %) improvement over PseudoSE. If both λs are non-zero, we see slight variations in the validation performance. When the noisy training data is non-purified, it is possible that the positive pair contrastive loss compels the model to enforce similarity on highly degraded pseudo-sources. These cases emphasizing premixture noise reconstruction similarity could cause the learned parameters to drift slightly away from speech-focused personalization.

The bottom subplot of Fig. 8 shows models pretraining through CM with data purification. Here, the most-optimal configuration is $\lambda_n = 0.001$ and $\lambda_p = 0.1$; the self-supervised model sees a 0.43 dB (or 6.6 %) improvement over PseudoSE. Notably, the positive pair-only models are able to obtain a 0.32 dB (or 4.9 %) improvement. With the CM loss functions weighted towards speech-dominant frames, we see that the positive and negative loss terms synergies more effectively.

One last observation is that the validation SDR of models using DP is overall lesser than that of models not using DP. This follows our hypothesis that the DP-based loss functions are more similar to the true fully-supervised speech enhancement loss. Note that all the self-supervised models are assessed on pseudo enhancement during validation. Therefore, it is understandable that the DP-based models have a lesser validation SDR improvement. The metrics computed at test-time assess true speech enhancement performance; therefore, observing this trend during validation alludes to greater enhancement.

Given our observation that CM works for many configurations, as a convenience for all other experiments, we set

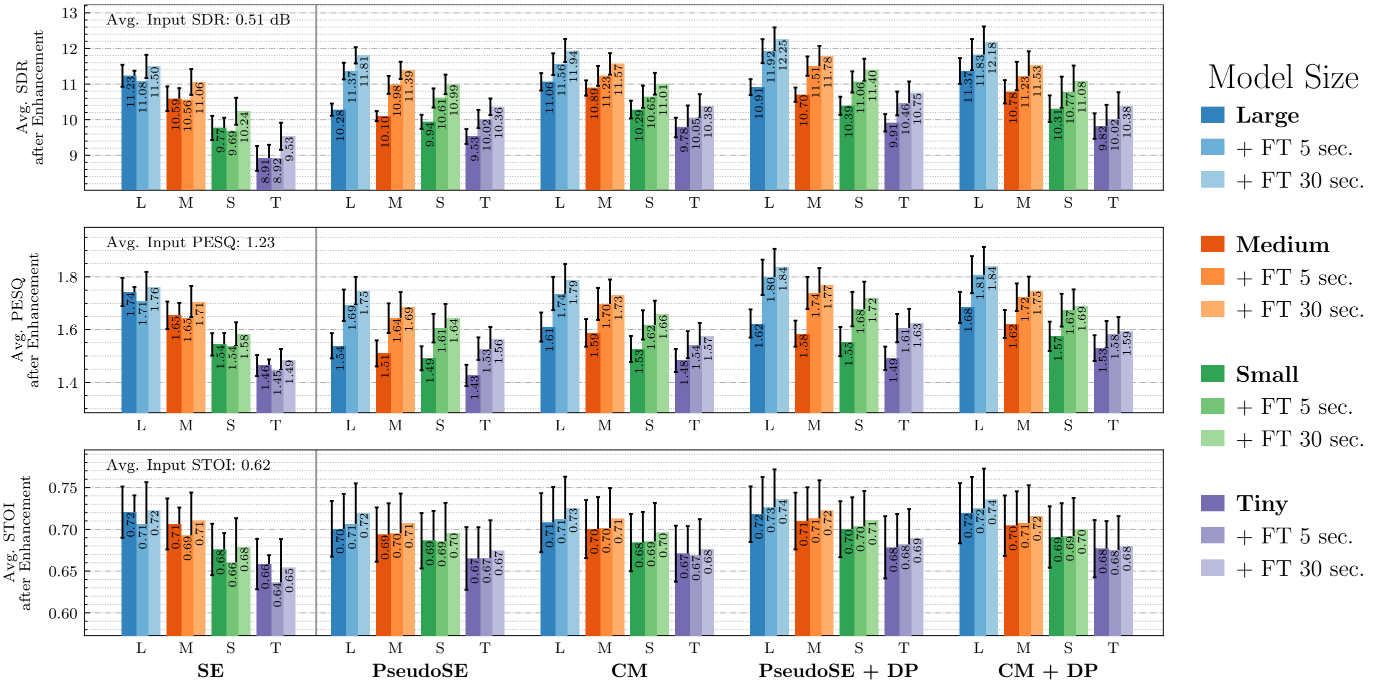


Fig. 9. Experiment results. The improvement for each metric (SDR, PESQ, and STOI) may be calculated by subtracting the average input value from the average value after enhancement. The plotted metrics are averaged over the 100 test set utterances for each of the 20 target speakers. The shading of each bar corresponds to the amount of clean speech data from the target speaker used for finetuning: 0 sec (i.e., no finetuning), 5 sec, and 30 sec. Error bars show the specific 95%-confidence interval per model and training configuration, averaged over all target speakers.

$\lambda_n = 0.1$ and $\lambda_p = 0.1$ with both non-purified and purified models.

B. *Efficient Personalized Speech Enhancement Study*

Next, we discuss the results from the main experiment. As described in in Sec. V, we consider 20 target speakers, 4 model sizes, 4 self-supervised pretraining methods, and 2 possible amounts of clean speech data. In terms of model checkpoints, there are 4 unadapted SE models, 160 fine-tuned SE models, 320 self-supervised PSE models, and 640 fine-tuned PSE models, resulting in a total of 1124 trials.

Figure 9 shows test set results in terms of the three signal quality metrics listed in Sec. V-B. Results are averaged across 20 test-time speakers, with different bars representing different model sizes and training configurations.

C. *ZSL Personalization Performance*

Bars with the darkest shading represent the performance of models in the ZSL personalization context, in which the models lack access to clean speech from the target speaker.

1) Generalist Models' Performance: The **SE** column's left-most bars show the performance of the bare generalist models' performance. The generalists are able to enhance the noisy test-time speakers in all cases, but it is clear that the larger models (bars labeled L or M) show much better generalization performance (up to 11.23dB SDR after enhancement) than the smaller ones (lower rows). For the tiny generalist models, the average SDR after enhancement is 8.92dB. This 2.31 dB range reinforces our argument that the smaller generalists tend to be poorer in generalization. Note that these baseline SE models are non-personalized. As they are without any adaptation, we can observe that the generalists' performance correlates with the architectural complexity because they are all trained using a large dataset.

2) Personalization using PseudoSE: The **PseudoSE** column shows the performance of the self-supervised models trained through pseudo enhancement of noisy speech targets. The model inputs are doubly-degraded observations of the test-time speaker ($\tilde{\mathbb{S}}_{\mathbf{p}\text{-tr}}$ is mixed with additional noise sources $\mathbb{N}_{\text{tr}}$), and the model naïvely recovers the pseudo-source. There is a chance that the pseudo targets are too far from clean speech, deviating the learned parametric function from the ideal personalized SE model. However, it is also possible that some parts of these pseudo speech sources are somewhat clean enough in order for the model to learn the target speaker's speech traits. The left-most bars (darkest shade) of the **PseudoSE** column do reveal success in personalization—note that the confidence interval of SDR enhancement narrows by using PseudoSE pretraining compared to SE pretraining. This trend is less obvious with perceptual metrics PESQ and STOI, but it is to be expected as the models' loss functions are SDR-based. PseudoSE does produce improvements over the **SE** pretraining when the models are tiny (9.53 vs. 8.91) or small (9.94 vs. 9.77). However, when the model complexity is large enough, we see that PseudoSE is unable to compete with the generalist model. Compare the largest model trained using PseudoSE against the largest speaker-agnostic SE model (10.28 vs. 11.23). Therefore, we conclude that PseudoSE's personalization performance is significant only when the model is incapable of learning from the large generic dataset.

3) Impact of DP with PseudoSE: As shown in our prior work [22], DP can identify cleaner frames from premixture signals $\tilde{\mathbb{S}}_{\mathbf{p}\text{-tr}}$ and improve the usability of the target speaker's noisy speech signals. We observe a similar trend with our ConvTasNet-based experiments. In particular, our results show that the **PseudoSE+DP** pretraining scheme in the ZSL context yields greater improvements over the plain **PseudoSE** in the large model than in the smaller ones. For example, introducing DP lifts the average performance of PseudoSE by $0.63\,\mathrm{dB}$ (10.91 vs. 10.28) in the large models, while the tiny models only see an average boost of about $0.38\,\mathrm{dB}$ (9.91 vs. 9.53). Because PseudoSE's efficacy is limited in the large models, the gains from introducing data purification are more prominent. However, it is still the case that the tiny model gains the most from the consolidated personalization process, e.g., a $1.0\mathrm{dB}$ improvement from the baseline SE model (9.91 vs. 8.91).

4) Personalization using CM: The ZSL results of the **CM** column are noteworthy because they compete with the **PseudoSE+DP** results despite using non-purified data. For example, **CM** results in better performance than **PseudoSE+DP** in large models (11.06 vs. 10.91) and works on par with **PseudoSE+DP** in small or tiny models. This shows that the proposed CM loss functions help the model learn robust features for personalized SE even though the signals used are noisy observations (or unlabeled in the sense of classification). These results validate the powerful feature learning capabilities of contrastive learning. Although the contrastive self-supervised learning paradigm has been explored in other research areas (e.g., SimCLR for computer vision), we note that the proposed CM pretraining method is specifically designed for source separation problems.

5) Impact of DP on CM: We find that **CM+DP** does not introduce significant improvements except with the largest model. This is likely due to the robust feature learning ability of CM, which is already competitive with the DP process.

6) Model Compression: Among the tiny-sized models, the best-performing ZSL method for personalization is **PseudoSE+DP** which produced an average SDR improvement of $9.91\mathrm{dB}$. We see that the personalized tiny model outperforms the generalist small model ($9.77\mathrm{dB}$), although it uses 62% fewer model parameters and multiply-accumulate operations (MACs) according to Table II. Likewise, the personalized small model comes within striking distance the medium-sized generalist (10.39 vs. 10.59) using less than 52% of the spatial and computational complexity. Finally, the best medium model after the **CM** personalization ($10.89\mathrm{dB}$) has its confidence interval overlapped with that of the largest SE baseline ($11.23\mathrm{dB}$), although its model complexity is less than 44%. From this we can conclude that, for lower-complexity models, the proposed self-supervised ZSL personalization may be viewed as a lossless model compression paradigm.

7) Success of Personalization: The height of the error bars indicate the 95%-confidence interval of each model and training configuration seen across the 20 target speakers. Using **SE** generalist pretraining, we observe that this variance can be as much as $0.9\,\mathrm{dB}$ for the tiny-sized models or $0.7\,\mathrm{dB}$ with the large-sized models. Through the proposed PseudoSE and CM methods, we see that the variance universally decreases in the ZSL context. Therefore, our self-supervised pretraining methods successfully adapt to the nuances of each test-time speaker despite being trained using only noisy data. Our results do show that introducing DP increases the variance in performance once again. This is to be expected as the availability of near-clean frames can differ greatly between speakers. Similarly, DP's reliance on the external SNR predictor model is also a contributing factor.

D. FSL Personalization Performance

Bars with lighter shading represent the FSL context, wherein models have $5\,\mathrm{sec}$ or $30\,\mathrm{sec}$ of clean speaker-specific data to finetune over.

1) Generalist Models' Performance: We observe that all four sizes of the baseline models pretrained as generalists (**SE**) are incapable of adapting over a small $\mathbb{S}_{\mathbf{f}\text{-tr}}$ that has only $5\,\mathrm{sec}$ of data. Using $30\,\mathrm{sec}$ of clean speech data does eventually produce gains for all model sizes. The tiny-sized generalist sees the most significant gains ($0.62\,\mathrm{dB}$) whereas the large-sized generalist sees marginal benefit ($0.27\,\mathrm{dB}$). This trend implies that the pretrained generalists are defined by model parameters that are too far from the ideally personalized counterpart, requiring much effort during the transfer learning process. In other words, too few clean utterances do not suffice in achieving the domain adaptation.

2) FSL after PseudoSE Initialization: We reiterate that our self-supervised methods train using noisy speaker-specific data with premixture SNRs in the $0\,\mathrm{dB}$ to $15\,\mathrm{dB}$ range. Hence, **PseudoSE** pretraining over this noisy data proves to be useful only for the tiny- and small-sized models (9.53 vs. 8.91 and 9.94 vs. 9.77), while the larger models do not benefit from the simple SSL setup. However, with all model sizes, finetuning using only $5\,\mathrm{sec}$ of clean data results in a significant performance boost (10.02 vs. 8.92, 10.61 vs. 9.69, 10.98 vs. 10.59, and 11.37 vs. 11.08).

Similar boosts also appear when using **PseudoSE+DP**, where all the performance scores are lifted by up to $0.84\,\mathrm{dB}$ (11.92 vs. 11.08 in the largest models). Our results suggest that finetuning is much more effective due to the speaker-specific self-supervised pretraining. By comparing the middle shaded bars in the **PseudoSE+DP** column with lightest shaded bars in the **SE** column, we can also see the data efficiency benefits of our self-supervised methods. In particular, after the **PseudoSE+DP** pretraining, only $5\,\mathrm{sec}$ of clean speech for finetuning achieve a greater mean SDR improvement compared to generalists models finetuned using $30\,\mathrm{sec}$ of clean speech. **PseudoSE+DP** achieves data efficiency with all model sizes (10.46 vs. 9.53, 11.06 vs. 10.24, 11.51 vs. 11.06, and 11.92 vs. 11.50). Our results show that through self-supervised pretraining, we are able to reduce reliance on the target speaker's private data by a factor of 6.

3) FSL after CM Initialization: In the ZSL context, **CM** pretraining produced notable improvements over **PseudoSE** likely due to the contrastive loss terms that introduce powerful regularization. But we found that the performance gap between CM and PseudoSE is nearly negligible in the FSL context. When it comes to data purification, we found that **CM+DP** was less effective in the FSL contexts than **PseudoSE+DP**. This is perhaps due to the data purification learning objective being too different from the contrastive learning objective, leading to a slightly sub-optimal joint learning objective. Nonetheless, for the ZSL scenario, CM pretraining without data purification has merit over PseudoSE, because it can alleviate the need for training a robust SNR predictor.

4) Model Compression: Finetuning also augments the model compression benefits of personalization. For example, we can use a small-sized **PseudoSE+DP** model finetuned with only $5\,\mathrm{sec}$ of clean speech to get $11.06\,\mathrm{dB}$ SDR after enhancement on average. This is on par with the largest **SE** model finetuned over the same amount of clean speech data ($11.08\,\mathrm{dB}$). This example shows a lossless 78% reduction in model parameters and MACs.

VII. Conclusion

We put forward self-supervised learning approaches towards personalized speech enhancement, highlighting their ability to learn robust features from the target speaker's noisy observations. Our main ideas are based on the assumption that noisy utterances of the target speaker might be more available than clean speech. However, due to the noisy nature of those unlabeled data, we propose more sophisticated SSL treatments to learn useful features from them. PseudoSE sets up a pretext SE problem where the enhancement target is still a noisy utterance. In addition, data purification improves the usability of the unlabeled (thus noisy) speech signals by identifying cleaner frames and focus more on them. With the purification step, PseudoSE becomes more realistic. Contrastive mixtures add an additional regularization benefit to the loss function, so that the pretext task is more relevant to the original source separation problem.

We observe that all these methods can act as a zero-shot personalization system which adapts to the target speaker's specificity with no additional clean speech used. In the few-shot learning context, we emphasize that the proposed SSL methods also serve as a better initialization scheme than a naïve generalist as the SSL methods learn from the target speaker's speech, even though it is contaminated. We found that the proposed systems quickly adapt using only a few seconds of test-user clean speech data, which is a too small amount for the baseline generalists to effectively perform transfer learning. Our results suggests that speaker-discriminative features can be found even in noisy recordings. The benefit of personalization is that it can reduce model complexity with no loss of SE performance, e.g., small personalized models perform as good as twice-larger general-purpose SE models. In addition, the proposed SSL methods make the few-shot learning-based personalization more data-efficient. Given that the transfer learning-based personalization requires clean speech data from the test-time users, reducing the required amount can improve the user experience.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 2046963.

References

[1] H. Taherian, Z.-Q. Wang, J. Chang, and D. 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.

[2] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[3] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "An Experimental Study on Speech Enhancement Based on Deep Neural Networks," *IEEE Signal Processing Letters*, vol. 21, no. 1, pp. 65–68, 2014.

[4] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Joint optimization of masks and deep recurrent neural networks for monaural source separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 12, pp. 2136–2147, Dec 2015.

[5] F. Weninger, H. Erdogan, S. Watanabe, E. Vincent, J. Le Roux, J. R. Hershey, and B. Schuller, "Speech Enhancement with LSTM Recurrent Neural Networks and its Application to Noise-Robust ASR," in *Proc. of the International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*, Aug. 2015.

[6] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep Clustering: Discriminative Embeddings for Segmentation and Separation," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2016.

[7] S. Pascual, A. Bonafonte, and J. Serra, "Segan: Speech enhancement generative adversarial network," *arXiv preprint arXiv:1703.09452*, 2017.

[8] H.-S. Choi, J.-H. Kim, J. Huh, A. Kim, J.-W. Ha, and K. Lee, "Phase-aware Speech Enhancement with Deep Complex U-Net," in *Proc. of the International Conference on Learning Representations (ICLR)*, 2018.

[9] K. Tan and D. Wang, "A Convolutional Recurrent Neural Network for Real-Time Speech Enhancement," in *Proc. of the Annual Conference of the International Speech Communication Association (Interspeech)*, 2018, pp. 3229–3233.

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

[11] D. L. Wang and J. Chen, "Supervised Speech Separation Based on Deep Learning: An Overview," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 10, pp. 1702–1726, 2018.

[12] J. Meyer, L. Rauchenstein, J. D. Eisenberg, and N. Howell, "Artie Bias Corpus: An Open Dataset for Detecting Demographic Bias in Speech Applications," in *Proceedings of the 12th Language Resources and Evaluation Conference*. European Language Resources Association, 2020, pp. 6462–6468.

[13] L. Deng and D. Yu, "Deep Learning: Methods and Applications," *Foundations and Trends in Machine Learning*, vol. 7, no. 3–4, pp. 197–387, 2014.

[14] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge Intelligence: Paving the Last Mile of Artificial Intelligence With Edge Computing," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1738–1762, 2019.

[15] Y. Jia, Y. Zhang, R. Weiss, Q. Wang, J. Shen, F. Ren, Z. Chen, P. Nguyen, R. Pang, I. L. Moreno, and Y. Wu, "Transfer Learning from Speaker Verification to Multispeaker Text-To-Speech Synthesis," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 31, 2018.

[16] C. Vondrick, A. Shrivastava, A. Fathi, S. Guadarrama, and K. Murphy, "Tracking Emerges by Colorizing Videos," in *Proc. of the European Conference on Computer Vision (ECCV)*, 2018.

[17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proc. of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, 2019, pp. 4171–4186.

[18] E. Jang, C. Devin, V. Vanhoucke, and S. Levine, "Grasp2Vec: Learning Object Representations from Self-Supervised Grasping," in *Proc. of The 2nd Conference on Robot Learning*, vol. 87, 2018, pp. 99–112.

[19] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A Simple Framework for Contrastive Learning of Visual Representations," in *Proc. of the International Conference on Machine Learning (ICML)*, 2020.

[20] Y. Gal, R. Islam, and Z. Ghahramani, "Deep bayesian active learning with image data," in *Proc. of the International Conference on Machine Learning (ICML)*, 2017, pp. 1183–1192.

[21] X. Dong and D. S. Williamson, "Long-term snr estimation using noise residuals and a two-stage deep-learning framework," in *Proc. of the International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*, 2018, pp. 351–360.

[22] A. Sivaraman, S. Kim, and M. Kim, "Personalized speech enhancement through self-supervised data augmentation and purification," in *Proc. of the Annual Conference of the International Speech Communication Association (Interspeech)*, 2021, pp. 2676–2680.

[23] Q. Wang, H. Muckenhirn, K. Wilson, P. Sridhar, Z. Wu, J. Hershey, R. A. Saurous, R. J. Weiss, Y. Jia, and I. L. Moreno, "Voicefilter: Targeted voice separation by speaker-conditioned spectrogram masking," *arXiv preprint arXiv:1810.04826*, 2018.

[24] R. Giri, S. Venkataramani, J.-M. Valin, U. Isik, and A. Krishnaswamy, "Personalized PercepNet: Real-Time, Low-Complexity Target Voice Separation and Enhancement," in *Proc. of the Annual Conference of the International Speech Communication Association (Interspeech)*, 2021, pp. 1124–1128.

[25] S. E. Eskimez, T. Yoshioka, H. Wang, X. Wang, Z. Chen, and X. Huang, "Personalized Speech Enhancement: New Models and Comprehensive Evaluation," *arXiv preprint arXiv:2110.09625*, 2021.

[26] M. Kolbæk, Z. H. Tan, and J. Jensen, "Speech Intelligibility Potential of General and Specialized Deep Neural Network Based Speech Enhancement Systems," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 1, pp. 153–167, Jan 2017.

[27] M. Kim, "Collaborative deep learning for speech enhancement: A run-time model selection method using autoencoders," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2017.

[28] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," in *Proc. of the International Conference on Learning Representations (ICLR)*, 2016.

[29] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[30] M. Kim and P. Smaragdis, "Bitwise neural networks for efficient single-channel source separation," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2018.

[31] L. Guo and M. Kim, "Bitwise source separation on hashed spectra: An efficient posterior estimation scheme using partial rank order metrics," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2018.

[32] S. Kim, M. Maity, and M. Kim, "Incremental binarization on recurrent neural networks for single-channel source separation," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2019.

[33] S. Kim, H. Yang, and M. Kim, "Boosted locality sensitive hashing: Discriminative binary codes for source separation," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2020.

[34] Y. Luo, C. Han, and N. Mesgarani, "Ultra-Lightweight Speech Separation via Group Communication," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 2021, pp. 16–20.

[35] A. Sivaraman and M. Kim, "Sparse Mixture of Local Experts for Efficient Speech Enhancement," in *Proc. of the Annual Conference of the International Speech Communication Association (Interspeech)*, 2020, pp. 4526–4530.

[36] R. E. Zezario, C.-S. Fuh, H.-M. Wang, and Y. Tsao, "Speech enhancement with zero-shot model selection," *arXiv preprint arXiv:2012.09359*, 2020.

[37] S. E. Chazan, J. Goldberger, and S. Gannot, "Speech Enhancement with Mixture of Deep Experts with Clean Clustering Pre-Training," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2021.

[38] A. Sivaraman and M. Kim, "Zero-shot personalized speech enhancement through speaker-informed model selection," in *Proc. of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2021.

[39] S. Kim and M. Kim, "Test-time adaptation toward personalized speech enhancement: Zero-shot learning with knowledge distillation," in *Proc. of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2021.

[40] S. Watanabe, T. Hori, J. L. Roux, and J. R. Hershey, "Student-Teacher Network Learning with Enhanced Features," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5275–5279.

[41] Y.-C. Wang, S. Venkataramani, and P. Smaragdis, "Self-supervised Learning for Speech Enhancement," *arXiv preprint arXiv:2006.10388*, 2020.

[42] M. Maciejewski, J. Shi, S. Watanabe, and S. Khudanpur, "Training Noisy Single-Channel Speech Separation with Noisy Oracle Sources: A Large Gap and a Small Step," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2021, pp. 5774–5778.

[43] T. Fujimura, Y. Koizumi, K. Yatabe, and R. Miyazaki, "Noisy-target Training: A Training Strategy for DNN-based Speech Enhancement without Clean Speech," in *Proc. of the European Signal Processing Conference (EUSIPCO)*, 2021, pp. 436–440.

[44] S. Wisdom, E. Tzinis, H. Erdogan, R. Weiss, K. Wilson, and J. Hershey, "Unsupervised Sound Separation Using Mixture Invariant Training," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[45] A. Sivaraman, S. Wisdom, H. Erdogan, and J. R. Hershey, "Adapting Speech Separation to Real-World Meetings Using Mixture Invariant Training," *arXiv preprint arXiv:2110.10739*, 2021.

[46] E. Vincent, C. Fevotte, and R. Gribonval, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

[47] K. Saito, S. Uhlich, G. Fabbro, and Y. Mitsufuji, "Training Speech Enhancement Systems with Noisy Speech Datasets," 2021.

[48] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[49] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "FSD50K: an Open Dataset of Human-Labeled Sound Events," *arXiv preprint arXiv:2010.00475*, 2020.

[50] D. Snyder, G. Chen, and D. Povey, "MUSAN: A Music, Speech, and Noise Corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[51] W. O. Olsen, "Average Speech Levels and Spectra in Various Speaking/Listening Conditions: A Summary of the Pearson, Bennett, & Fidell (1977) Report," *American Journal of Audiology*, vol. 7, no. 2, pp. 21S–25, 1998.

[52] K. Smeds, F. Wolters, and M. Rung, "Estimation of Signal-to-Noise Ratios in Realistic Sound Scenarios," *Journal of the American Academy of Audiology*, vol. 26, no. 02, pp. 183–196, 2015.

[53] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio Set: An Ontology and Human-Labeled Dataset for Audio Events," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2017.

[54] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 2, 2001, pp. 749–752.

[55] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "A short-time objective intelligibility measure for time-frequency weighted noisy speech," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2010.

[56] K. Yoshii, R. Tomioka, D. Mochihashi, and M. Goto, "Beyond NMF: Time-Domain Audio Source Separation without Phase Reconstruction," in *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, 2013, pp. 369–374.

[57] S. Venkataramani, J. Casebeer, and P. Smaragdis, "End-To-End Source Separation with Adaptive Front-Ends," in *Proc. of the Asilomar Conference*. IEEE, 2018, pp. 684–688.

[58] D. Stoller, S. Ewert, and S. Dixon, "Wave-U-Net: A Multi-Scale Neural Network for End-to-End Audio Source Separation," in *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, 2018, pp. 334–340.

[59] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," pp. 4510–4520, 2018.

[60] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019, pp. 8024–8035.

[61] M. Pariente, S. Cornell, J. Cosentino, S. Sivasankaran, E. Tzinis, J. Heitkaemper, M. Olvera, F.-R. Stöter, M. Hu, J. M. Martín-Doñas, D. Ditter, A. Frank, A. Deleforge, and E. Vincent, "Asteroid: The PyTorch-Based Audio Source Separation Toolkit for Researchers," in *Proc. of*

the Annual Conference of the International Speech Communication Association (Interspeech), 2020.

[62] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. of the International Conference on Learning Representations (ICLR)*, 2015.

VIII. Biography Section

Aswin Sivaraman is currently a Ph.D candidate at Indiana University in the Department of Intelligent Systems Engineering. He received his BS degree in electrical engineering from the University of Illinois at Urbana-Champaign in 2015. Aswin was previously with Qualcomm as a software engineer, and has been an intern for Google, X, Spotify, and Amazon. He is a recipient of the 2017 Graduate Fellowship at Indiana University and the Interspeech 2020 Student Travel Grant. His research focuses on deep learning methods for speech and music processing.

Minje Kim (M'12–SM'19) is Assistant Professor of Intelligent Systems Engineering at Indiana University, where he is also affiliated with Data Science, Cognitive Science, and Statistics. He is also an Amazon Visiting Academic. He received the Ph.D. degree in Computer Science from the University of Illinois at Urbana-Champaign in 2016. Before that, he worked as a researcher at ETRI, Daejeon, Korea from 2006 to 2011. He is an IEEE Senior Member and a member of the IEEE AASP TC. He is a recipient of the NSF CAREER Award (2021), IEEE SPS Best Paper Award (2020), Google and Starkey's grants for outstanding student papers at ICASSP 2013 and 2014, respectively. His research spans machine learning and audio signal processing.