Zero-shot test-time adaptation via knowledge distillation for personalized speech denoising and dereverberation

Sunwoo Kim; Mrudula Athi; Guangji Shi; Minje Kim; Trausti Kristjansson

https://doi.org/10.1121/10.0024621
Zero-shot test-time adaptation via knowledge distillation for personalized speech denoising and dereverberation

Sunwoo Kim,1 Mrudula Athi,1 Guangji Shi,1 Minje Kim1,2,a) and Trausti Kristjansson1
1Amazon Lab126, Sunnyvale, California 94089, USA
2Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA

ABSTRACT:
The proposed zero-shot personalization method uses no clean speech target. Instead, it employs the knowledge distillation framework, where the more advanced denoising results from an overly large teacher work as pseudo targets to train a small student model. Evaluation on various test time conditions suggests that the proposed personalization approach can significantly enhance the compact student model’s test time performance. Personalized models outperform larger non-personalized baseline models, demonstrating that personalization achieves model compression with no loss in dereverberation and denoising performance. © 2024 Acoustical Society of America.
https://doi.org/10.1121/10.0024621

Effect: B. Yegnanarayana
Pages: 1353–1367

I. INTRODUCTION

Real-world speech signals are often corrupted by a varying level of interfering noise and reverberation, which can be detrimental to the performance of audio applications. Hence, speech enhancement (SE) algorithms are an essential component incorporated into the audio applications such as automatic speech recognition, diarization, voice-over-IP, and transcription (Boll, 1979; Ephraim and Malah, 1984; Gannot et al., 1998). Among these potential applications, in this paper, we focus on the direct use of the enhanced speech for voice communication, i.e., improving the perceptual speech quality of the end user is our goal.

Recent advancements with deep neural networks (DNNs) for SE have shown superior performance compared to traditional machine learning and signal processing methods (Chazan et al., 2017; Wang and Chen, 2018; Xu et al., 2014). However, these state-of-the-art applications require significant memory and computational bandwidth, rendering them difficult for deployment onto devices for practical uses. Resource-constrained devices, such as hearing aids or wearable devices, cannot efficiently handle real-time inference tasks for the SE applications when the models are with multilayered complex architectures.

Consequently, research in model compression methods have gained interest to address the practicality of deep learning architectures for real-time applications. Common compression methods include quantization to reduce the bit-width (Bhalgat et al., 2020; Jacob et al., 2018), pruning to induce sparsity in model weights (Frankle and Carbin, 2018; Han et al., 2016), and knowledge distillation (KD) to distill larger teacher models onto smaller student models (Hinton et al., 2015). These methods, such as quantization, pruning, and KD, have shown great promise in reducing the model size and complexity while minimizing the drop in generalization performance. However, this kind of compression method can be seen as context-agnostic as they do not use the specificity of the test time context. Instead, they tend to seek a general-purpose compression technique that works reasonably well in various real-world test conditions. As a result, a certain level of performance drop is inevitable after compression.

In this paper, we aim at developing a context-aware DNN compression method for SE. We envision that a compressed model can reduce its run-time complexity without losing its performance if it focuses on a particular test environment. We contrast the proposed concept and the ordinary DNN-based SE models, which are typically designed as general-purpose frameworks with a large architecture. A DNN’s large capacity is fully employed when it is trained on a large training set that consists of various speakers and noise sources, generalizing well to unseen test time conditions, e.g., different speakers, noise sources, signal-to-noise ratios (SNRs), and room acoustics. In some practical use-cases,
though, it suffices for the enhancement model to perform well only for the specific test time context. For instance, a family-owned smart assistant device sitting in the living room needs to perform well only for the family members’ voices and their home acoustics but not necessarily for the other situations. Compared to the general-purpose SE model, the generalist, our context-aware compression method can allow a model to adapt to the specific speakers and their acoustic context, overcoming the generalization losses. We call this type of context-aware SE model a personalized speech enhancement (PSE) system. Because the test time context contains limited variability, a small personalized model can even outperform larger and more complex universal generalist models, demonstrating personalization as a form of model compression.

The proposed PSE models achieve context-awareness by reducing the domain mismatch between the training and test datasets. The topic of domain adaptation has been an active area of research in machine learning. One common procedure for domain transfer is regularizing the differences between the learned representations of source and target datasets. It has been applied for emotion, speech, and speaker recognition (Deng et al., 2014; Sun et al., 2017). However, these applications rely on ample target data, which cannot be assumed if the target problem is narrowly defined as in our PSE cases. Few-shot adaptation can be a solution as it requires only a small amount of ground-truth signal (Sivaraman and Kim, 2022). However, it can be challenging to obtain ground-truth user information due to recent privacy infringement, data leakage issues. For example, the advancement in DeepFake technology increased people’s concern toward releasing personal information (Rochford, 2018). Meanwhile, with user compliance, the user enrollment phase can obtain trigger phrases from the users, but there is no guarantee that these recordings are clean enough to be used as the target of PSE.

In contrast to aforementioned approaches, zero-shot learning is a solution suitable for training tasks when no additional labeled data are available (Wang et al., 2019; Xian et al., 2018). In the context of personalization, a zero-shot approach means that it does not require test users’ clean speech data or their home acoustic environment while its goal is still to adapt to the test time specificity. Zero-shot learning is an active research topic for classification tasks, where test time prediction is performed by computing auxiliary information (Larochelle et al., 2008; Palatucci et al., 2009; Romera-Paredes and Torr, 2015). Similarly, zero-shot learning in the speech and audio classification applications extracts semantic properties or articulatory distribution to obtain labels during test time (Choi et al., 2019; Dauphin et al., 2013; Li et al., 2020).

However, zero-shot learning for SE has not been widely studied yet. In Sivaraman and Kim (2020, 2021), a mixture of local expert model is introduced as a zero-shot solution to test time adaptation of a SE model. It achieves the adaptation goal by employing an internal classifier to select the most suitable one out of predefined specialist models for a given noisy test signal, selecting a predefined specialist model for a given noisy test signal. Although it is a valid adaptation method, it only works on a few predefined contexts, i.e., varying input SNR levels, the gender of the speakers, and predefined speaker groups, rather than actively learning from the test time speaker’s personality or the unique context. In Sivaraman et al. (2021), self-supervised learning methods are proposed to achieve PSE, where a data purification algorithm identifies clean speech frames from test time noisy speech. Although it achieves the PSE goals, it is not fully using the test time observations, which can be rare. Moreover, the data purification method does not apply to the dereverberation problem. Other works introduce a zero-shot solution for deep clustering-based speech separation models to estimate absent ground-truth labels. Provided a multichannel input, the affinity matrices were estimated through unsupervised spatial clustering or by using phase difference features (Drude et al., 2019; Tzinis et al., 2019). However, as a result of the sheer size and inference costs, deep clustering models are difficult to fit on small devices. In addition, these models are typically for speech separation problems rather than SE.

In this paper, we present a zero-shot learning approach to personalization for joint dereverberation and denoising based on the KD framework (Hinton et al., 2015). As a zero-shot learning method, it does not ask for clean ground-truth signals from the user while it still aims at enhancing noisy reverberant mixtures. Because its goal is to train a small specialist model for a particular user’s speech and recording environment, it qualifies as a personalization method. For zero-shot learning approaches implemented via student-teacher strategies, it is common to use data synthesis techniques through generative adversarial frameworks, where the generator produces fake samples (Chen et al., 2019; Micaelli and Storkey, 2019; Ye et al., 2020). Pseudo samples can be also synthesized using activation or output statistics of trained teacher models (Lopes et al., 2017; Nayak et al., 2019) among many others (Chawla et al., 2021; Nayak et al., 2021). Instead of including an intermediate data synthesis step, our proposed model directly uses the teacher model’s outputs as if they were ground-truth targets. Previous works have successfully applied KD to develop compact SE systems (Hao et al., 2020; Nakaoka et al., 2021; Subramanian et al., 2018; Tu et al., 2019). Under the KD framework, the basic assumption is that the teacher model’s large computational capacity guarantees the generalization goal. We extend this concept to a novel zero-shot learning approach for PSE. As the teacher model works well in most test time environments, we consider its excellent SE results as if they were the target clean speech from the student model’s perspective. That way, we can turn any noisy and reverberant test signals into labeled training examples by passing them through the teacher model. In this process, the teacher model remains as a generalist model, whereas the student model can use the teacher’s generalization power to learn from the test time input signals, fulfilling the zero-shot learning condition.
Using a KD learning paradigm enables us to leverage noisy unlabeled data and obtain their corresponding soft targets generated by the teacher model. In this paper, we focus on domain adaptation when we have large unlabeled target-domain dataset and assume noisy speech data of a target test time user to be more widely available as opposed to their clean labeled data. Under this assumption, we approach the PSE problem with a self-supervised method in which corresponding pseudo targets are generated from large amounts of unpaired noisy speech data (Doersch et al., 2015; Manohar et al., 2018; Watanabe et al., 2017; Zhang et al., 2020). Our experiments show that the small student models can be personalized in this way, resulting in improved performance compared to their context-agnostic counterparts. Moreover, given that these models are still small, performance improvement reconstitutes the whole KD process as a model compression method. For example, our experiments consistently show that the PSE models can compete with their larger generalist counterparts. We envision that the compact student models can work as an affordable solution in edge devices with limited computing resources.

Figure 1 provides an overview of the proposed KD-based PSE process. On the left, as a pretraining step, the teacher and student models are trained from a generic dataset to cover all test time variations. However, the student model’s constrained capacity tends to limit its SE performance. At the center, KD-based fine-tuning learns from the target test environment: the estimated clean speech by the student model is compared against the result from a larger teacher model, whose discrepancy is used to fine-tune the student model. The zero-shot framework enables test time adaptation.

When deploying our framework, we ultimately use only the student model on the device for the PSE inference (the rightmost part of Fig. 1). Even after being deployed, this student model can continue to be refined: the device collects more contaminated speech signals from the test scene, which are then fed to the teacher model to produce the corresponding pseudo clean speech target. The pairs of newly generated target speech signals are then fed to the teacher model to produce the corresponding pseudo clean speech target. The zero-shot framework enables test time adaptation.

FIG. 1. (Color online) An overview of the proposed KD-based PSE process shows (left) the pretraining process for teacher and student models using generic dataset, (center) the KD-based personalization process, and (right) the student model’s inference process after the personalization.
the effects of personalization to various unseen environments. Concluding remarks are presented in Sec. V.

II. THE PROPOSED KD-BASED ZERO-SHOT PSE ALGORITHM

Given a monaural signal recorded in a noisy and reverberant environment, we formulate the signal model as

\[ y[t] = x[t] + zn[t] = s[t] * h[t] + zn[t], \]  

(1)

where \( s, n, h, \) and \( x \) denote speech source, background noise, room impulse response (RIR) function, and reverberant speech, respectively. The symbol “*” stands for the convolution operator. The parameter, \( x \), controls the SNR between the reverberant speech and interfering noise source. Our goal in this study is to recover the clean anechoic version of the single-talker speech signal, \( s \), from the corresponding noisy reverberant observation, \( y \).

We propose a KD-based zero-shot PSE algorithm, which aims at joint denoising and dereverberation. Our goal is to fine-tune a compact student model after it is deployed such that it adapts to the unseen test speaker and environment continuously. In doing so, the teacher model’s powerful generalization performance plays a significant role as it performs denoising and dereverberation simultaneously, a behavior that the student model attempts to learn from.

A. Training teacher SE models

First, we train the teacher model, \( T(\cdot) \), using a large-scale dataset consisting of dry speech sources, various noise signals, and RIR filters. Here, \( T(\cdot) \) is defined with a large model architecture, therefore, it can properly approximate the complex general-purpose joint speech denoising and dereverberation function. Once trained, \( T(\cdot) \) is frozen and not fine-tuned, assuming that its SE performance as a generalist meets the quality standard in most test cases. Another assumption is that it is too complex for the given test time user device to perform real-time SE inference tasks.

To train the teacher models, we use generic training datasets. The formulation of the training dataset is as follows. The clean speech utterances are taken from a large corpus containing many speakers, \( s \in S \); the noise recordings are also from a large corpus containing various noise types, \( n \in N \); the RIRs are similarly from a large collection recorded in various rooms, \( h \in \mathbb{H} \). We use them to synthesize the noisy and reverberant signals, \( y \), as input [Eq. (1)].

Hence, the goal of the teacher model is to jointly denoise and dereverberate \( y \) so the model can estimate the waveforms, \( \hat{s} \), that closely approximate the target clean anechoic speech, i.e., \( s \approx \hat{s} = T(y) \). The optimization on \( T(\cdot) \) reduces the loss between the target utterance, \( s \), and reconstruction, \( \hat{s} \), i.e.,

\[ \arg\min_{\Theta_T} L(s || T(y; \Theta_T)), \]  

where \( \Theta_T \) denotes the trainable parameters of the teacher model. Note that the training process for the teacher model corresponds to the typical supervised learning method for general-purpose SE. Detailed model and optimization descriptions are provided in Sec. III B.

B. Pretraining student SE models

Our student models, \( S(\cdot) \), are pretrained in a similar way to the teacher models, i.e., by updating its own model parameters, \( \arg\min_{\Theta_S} L(s || S(y; \Theta_S)) \), using the same generic datasets, \( G, N, \) and \( H \). However, its small capacity hinders it from generalizing well to the unseen test conditions. Thus, we argue that further improvement is required for these student models to meet the quality requirement. We introduce the KD-based test time personalization algorithm in Sec. II C, which is designed to reduce the performance gap between \( T(\cdot) \) and \( S(\cdot) \). In this regard, the purpose of pretraining \( S(\cdot) \) is to prepare the student model better than a random initialization, primed for the next fine-tuning step. Further details on model and training are also given in Sec. III B.

C. Test time PSE

During the test time, we assume that the enhancement system is exposed to mixture signals composed of clean speech utterances from the test speaker, \( s \in S \), background noise sources, \( n \in M \), and RIRs, \( h \in R \). Note that we differentiate these test sets from the training sets, i.e., \( G \neq S, N \neq M, \) and \( H \neq R \). Meanwhile, we also assume that the noisy and reverberant speech signals defined by the combination of all speech, noise, and RIRs available in the training sets \( G \times N \times H \) mixed through Eq. (1) are representative enough to encompass the test time variations, i.e., \( S \times M \times R \subseteq G \times N \times H \). In practice, however, there might be corner cases that even the large dataset \( G \times N \times H \) cannot successfully represent, which the proposed method could fail to adapt to. Hence, we postulate that if a teacher model is large enough, it can serve as an unbiased solution to the denoising and dereverberation problem. Meanwhile, a small student model is also in our interest if it is small enough for the resource-constrained edge device. However, it may be too biased to generalize well to the test time SE task because of its small model capacity.

Given these assumptions, we propose a personalization framework that can adapt to a new environment without requiring test user’s ground-truth clean speech samples or any other auxiliary information of the speakers and acoustic scene. As we formulate the proposed personalization method as a fine-tuning process, we begin with a compact student model, \( S(\cdot) \), pretrained in a context-agnostic manner as in Sec. II B. To fine-tune the student model, its enhancement result from dereverberation and denoising, \( \hat{s}_S \), must be compared against the target to compute the loss and perform backpropagation. However, because we assume that the target is not available, we use the pseudo target computed from the teacher model.

This process falls in the category of the student-teacher framework in which a student model is optimized using a teacher model’s prediction (Hinton et al., 2015). In the context of PSE, we employ a large pretrained teacher model, \( T(\cdot) \),
whose predicted clean utterance serves as the target to compute the student model’s loss. Student and teacher models are initialized with pretrained generic enhancement models as discussed in Secs. II A and II B, respectively. During the test time, the student model is optimized as \( \arg\min_{\theta_S} \mathcal{L}(\mathcal{T}(\varepsilon^s), \mathcal{S}(y, \Theta^s)) \), where \( \mathcal{T} \) are the estimates of clean speech signals obtained from the teacher model, and \( \Theta^s \) are trainable parameters of the student model. We distinguish this fine-tuned student model, \( \mathcal{S}(\cdot) \), from the pretrained model, \( \mathcal{S}(\cdot) \), from now on.

The teacher’s estimate, \( \mathcal{T} \), is only an approximation of the ground-truth target, \( y \), and can contain artifacts from dereverberation and denoising (Xu et al., 2014). However, under a zero-shot PSE setup, we assume that having these synthesized pseudo targets is better than nothing. Hence, the performance of the fine-tuning results depends on the quality of \( \mathcal{T} \). To this end, we employ relatively large models that surely outperform the student models on the test signals, i.e., \( \mathcal{L}(y, \mathcal{T}(\varepsilon^s)) \). Given its large capacity, the teacher model generalizes better to unseen inputs compared to the student model. Thus, we hypothesize that the student still learns from these imperfect targets and adapts to the test environment. In our experiments on simulated signals and real-world test environments, we show that this assumed performance gap exists, guaranteeing the performance improvement by the KD process.

D. Interpretation from a manifold assumption

By training under the SE criterion, models learn to produce latent representations that are robust to corruptions in input data and useful for recovering the clean speech. Successfully learned latent representations are discriminative and can capture useful structure and variations in the input distribution as discussed in the context of denoising autoencoders (Vincent et al., 2010). We interpret the process of SE using the manifold assumption: high dimensional clean speech data lie on a low dimensional manifold (Chapelle et al., 2006). Data samples are mapped onto a manifold that represents a feature space that preserves the local structure of the data. The objective of our models is to learn the underlying manifold of the speech signals such that they can accurately map the noisy samples to their respective positions on the manifold of clean speech during test time. In Fig. 2, we see generic clean speech samples, \( s \) (the crosses), form a complex manifold (the thin solid line). Meanwhile, under our PSE assumption that test time environments will contain smaller subset of sources (e.g., speakers, noises, and room variation), the models would only need to learn the manifold of those subsets, which imply a simpler manifold (the thick solid line) than the generic speech’s. Under this interpretation, the target speaker’s corrupt examples are mapped away from the manifold of clean signals. SE models try to project the off-the-manifold examples, \( y \), back onto the manifold. The farther away \( y \) is from the manifold, the more corrupt the example is, and the model takes bigger efforts to reach the manifold. Note that the corrupted samples, \( y \), are spoken by the same target person and, therefore, the target manifold is the simpler manifold (thick line) than the complex manifold (the thin solid line) by all people.

We expect a large complex model, \( T(y) \), to better approximate the manifold given its larger architecture. Hence, its prediction of the personal clean speech forms an approximation (thin dotted line) similar to the original approximation (the thick solid line). On the contrary, smaller models are likely to learn a poor approximation, \( \mathcal{T} \) (the thin dash). The aim of our proposed personalization framework is to distill the better manifold determined by \( \mathcal{T} \) to the smaller student models to help better approximate the manifold. By doing so, student models fine-tuned under our framework will be able to better define and map points, \( y \), closer to the test time manifold, which is approximated by \( \mathcal{T} \) (the thick dash).

III. EXPERIMENTAL SETUP

A. Datasets

Table I summarizes the datasets that were used for the experiments. For pretraining, we employed clean speech recordings from the LibriSpeech corpus (Panayotov et al., 2015) and noise recordings from the MUSAN (Snyder et al., 2015) and ESC50 datasets (Piczak, 2015). For RIRs, we used publicly available recordings downloaded using Kaldi scripts. The RIR data sources consist of the Aachen Impulse Response Database (Jeub et al., 2009), PORI

![Manifold learning perspective of SE, considering small model sizes and the potential subsampling of the dataset to construct personalized dataset and model.](https://doi.org/10.1121/10.0024621)

**FIG. 2.** (Color online) Manifold learning perspective of SE, considering small model sizes and the potential subsampling of the dataset to construct personalized dataset and model.

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speech</td>
<td>Librispeech (Panayotov et al., 2015)</td>
</tr>
<tr>
<td>Noise</td>
<td>MUSAN (Snyder et al., 2015)</td>
</tr>
<tr>
<td>Noise</td>
<td>ESC50 (Piczak, 2015)</td>
</tr>
<tr>
<td>RIR</td>
<td>AIR (Jeub et al., 2009)</td>
</tr>
<tr>
<td>RIR</td>
<td>PORI (Merimaa et al., 2005)</td>
</tr>
<tr>
<td>RIR</td>
<td>RWCP (Nakamura et al., 2000)</td>
</tr>
<tr>
<td>RIR</td>
<td>BUT (Szöke et al., 2019)</td>
</tr>
<tr>
<td>RIR</td>
<td>REVERB (Kinoshita et al., 2013)</td>
</tr>
</tbody>
</table>

TABLE I. Corpus and notation of speech, noise, and RIR datasets used during pretraining and personalization.
concent hall impulse responses (Merimaa et al., 2005), and RWCP Sound Scene Database in Real Acoustical Environments (Nakamura et al., 2000). We used Librispeech’s train-clean-360, MUSAN’s free-sound, and the collective RIR data for training, which we denote as G, N, and H, respectively. A comprehensive summary of RIR datasets, including information on RT60, number of rooms, and microphone to loudspeaker distance, can be found in Merimaa et al. (2005) and Szöke et al. (2019). This exposes the generalist models to up to 251 speakers, 843 noise recordings, and 334 RIRs during training. The noisy mixtures are obtained by adding the noise to speech signals at random input SNR levels uniformly chosen between $-5$ and $10$ dB.

For fine-tuning or zero-shot PSE, we used 44 speakers from Librispeech’s test-clean and noise from the ESC-50 dataset for environmental sound classification with 50 different noise types from 5 categories, consisting of animals, natural and water soundscape, nonspeech human sounds, interior-domestic sounds, and exterior-urban sounds. For RIRs, we used 5 rooms from BUT Speech@FIT Reverb Database (BUT; Szöke et al., 2019) and real rooms from the Reverb 2014 Challenge (RVB) dataset (Kinoshita et al., 2013) for a total of 11 rooms. Each room in the BUT dataset contains 31 microphones and 5 source positions on average. RIRs were measured for each speaker position using exponential sine sweep method. For RVB dataset, there are three types of rooms (small, medium, and large) and two types of microphone placement (near and far). RIRs are collected from two microphone angles per room. Further information on RIR datasets, along with speech and noise corpuses, can be found in Table I (Szöke et al., 2019).

We synthesize $K = 44$ unique test time environments, each of which consists of a test speaker, a noise source, and a RIR configuration defined by the location of the speaker and microphone. In particular, given a test environment index, $k \in \{1, \ldots, K\}$, we sample clean utterances from the $k$th speaker, $S^{(k)}$, convolve it with the $k$th room’s RIR, $R^{(k)}$, and add noises from the $k$th noise type, $M^{(k)}$. For each test environment, $S^{(k)}$ are split into separate sets for fine-tuning, validation, and testing: the partitions are approximately 5, 1, and 1 min of clean speech, which we denote by $S^{(k)}_{te}$, $S^{(k)}_{va}$, and $S^{(k)}_{te}$, respectively. The noise and RIR samples are prepared similarly and partitioned into three separate sets. We synthesize noisy and reverberant input signals by combining $S^{(k)}_{te}$, $M^{(k)}$, and $R^{(k)}_{te}$. Haviing them as input, the student model is fine-tuned via the KD process, where the teacher model’s denoising results are used as the pseudo target. In other words, the student model for generic SE is first deployed to the device and can be personalized to the user’s specificity using 5 min of noisy and reverberant recordings of the test environment. Then, $S^{(k)}_{va}$, $M^{(k)}$, and $R^{(k)}_{va}$ are used to validate the student model during fine-tuning, mainly to prevent overfitting. Note that this does not mean that the PSE algorithm needs clean speech for validation: the validation process still relies on the teacher’s estimate of clean speech as the target to compute the validation loss. Hence, early stopping is still conducted in a zero-shot manner. We report our PSE models’ final performance using the test sets $S^{(k)}_{te}$, $M^{(k)}_{te}$, and $R^{(k)}_{te}$, for which we do compute the final enhancement performance by comparing to the ground-truth clean speech signals, $S^{(k)}_{te}$.

When we simulate various test conditions, the noise and speech sources are mixed under four different input SNR levels (i.e., $-$5 dB, 0 dB, 5 dB, and 10 dB). All speech and noise audio files are loaded at 16 kHz sampling rate and standardized to have unit-variance.

B. Models

Our student models are based on the unidirectional gated recurrent unit (GRU) architecture (Cho et al., 2014). Recurrent neural networks with gating technology, such as the long short-term memory (LSTM) cell (Hochreiter and Schmidhuber, 1997) and GRUs, have been predominantly used in SE due to their ability to overcome the gradient vanishing or explosion issues during backpropagation through time (BPTT). As for GRUs, although it was first introduced as a computationally efficient alternative of LSTM for machine translation, it was quickly adopted for SE tasks as a result of their flexibility in handling continuous input sequences (e.g., audio spectra) and learning continuous latent variables (Chazan et al., 2017; Luo et al., 2020; Sivaraman and Kim, 2020). In the SE literature, LSTM and GRU can be combined with Convolutional Neural Network (CNN) layers for better performance (Hu et al., 2020), but the hybrid architecture adds more burden to the hardware design. In this paper, we focus on the simple GRU-only architecture, which is more suitable for CPU operations, and show the PSE method’s merits. However, the proposed principles should apply to other architectural choices.

We use frequency-domain representations obtained through the short-time Fourier transform (STFT) as inputs to the enhancement models. STFT is with a Hann windowed frame of 1024 samples and a hop size of 256 samples. The recurrent unit reads each STFT magnitude spectrum sequentially and updates the hidden state at each frame. For our denoising application, we apply a dense layer to map the hidden unit outputs from the GRU layer into complex ideal ratio masks (Williamson et al., 2016). The denoising mask is applied element-wise to the mixture complex spectrogram, and then transformed back to the time-domain signal, $\hat{s}$, through inverse STFT. We use negative scale-invariant signal-to-noise ratio (SI-SNR) as the loss function (Le Roux et al., 2019). While the GRU architecture for the student models is fixed with two hidden layers, we vary their hidden units from 32 to 1024 to verify the impact of personalization on the different architectural choices.

Meanwhile, as for the teacher model, we employ two different network architectures. First, we use a $3 \times 1024$ GRU architecture, which is large enough to outperform the students. In addition, we also employ dual-path Recurrent Neural Networks (RNN) (DPRNN; Luo et al., 2020) as an alternative teacher model. DPRNN was chosen because of
its higher performance and smaller model size compared to other time-domain models such as Conv-TasNet (Luo and Mesgarani, 2019). More advanced transformer based models, such as dual-path transformer (DPTNet; Chen et al., 2020), report higher performance with comparative size to DPRNN in speech separation tasks, but we found, empirically, that the DPRNN performs better for our dereverberation and denoising task.

Indeed, the DPRNN teacher outperforms the GRU teacher because of its structural advantage. Hence, we contrast the impact of the two teacher models on the PSE performance after the KD-based fine-tuning process. The DPRNN model is configured using implementation available in Asteroid’s source separation toolkit (Pariente et al., 2020). Same architecture as was reported in Luo et al. (2020) is adopted (i.e., six repeats), and we trained it with our single-speaker SE setup rather than the original speech separation task. The model architectures, their respective number of parameters, and the multiplier-accumulator (MAC) operation counts are shown in Table II. Note that DPRNN is not the largest model, but it requires extensive MAC operations.

Here, we introduce new notations to distinguish between the two teacher model architectures: $T_{\text{GRU}}$ and $T_{\text{DPRNN}}$. In addition, we also denote the fine-tuned student models differently from the pretrained initial model, $S$, and add the subscript to indicate what it learns from: $S_{\text{GRU}}$ and $S_{\text{DPRNN}}$, respectively. We include a student model fine-tuned on the ground-truth oracle clean speech targets as a performance upper bound and denote it as $S_{\text{GT}}$. Summary of notations for pretrained and fine-tuned models are listed in Table III. Note that the systems denoted with tilde, $S_{\text{GRU}}$, $S_{\text{DPRNN}}$, and $S_{\text{GT}}$, represent personalized student models, whereas the generalist models, $S$, $T_{\text{GRU}}$, and $T_{\text{DPRNN}}$, are discussed to report the performance of either the teacher models or the baseline.

The Adam optimizer (Kingma and Ba, 2015) was used with learning rate of $1 \times 10^{-4}$ for pretraining and $1 \times 10^{-5}$ for fine-tuning.

### IV. EXPERIMENTAL RESULTS AND DISCUSSION

#### A. Objective performance analyses

The box plots in Fig. 3 show the enhancement performances of various models under $K = 44$ environments synthesized with different noise level conditions. The results are shown for pretrained and fine-tuned student models as well as for teacher models as reference. Figures 3(a)–3(d) show comprehensive denoising and dereverberation results across cases with severe (−5dB) to moderate (−10dB) test time SNR levels. We notice that the overall performance decreases due to the additive background noise. While their final input SNR values are controlled by varying the loudness of the test noise sources, $M^{(t)}_{\text{ne}}$, the speech sources are also degraded by the reverberation defined by the test time RIR set, $B^{(t)}_{\text{rg}}$. In Figs. 3(a)–3(d), we observed that our proposed personalization framework improves dereverberation and denoising performances of pretrained student models under all noise and room conditions, i.e., $S_{\text{GRU}}$ and $S_{\text{DPRNN}}$ results are always better than the $S$ results, on average, if their model complexity is the same. From these results, we can infer that personalization helps significantly improve the joint dereverberation and denoising performances for all student architectures.

In addition, we observe that the personalized models learned from the DPRNN teacher, $S_{\text{DPRNN}}$, always outperform their corresponding models fine-tuned using the GRU teacher, $S_{\text{GRU}}$. $S_{\text{GRU}}$, at times, perform similarly to $T_{\text{GRU}}$ especially in the case of the $2 \times 1024$ and $3 \times 1024$ student and teacher models. This is because both models trained under the same complex Ideal Ratio Mask (cIRM) estimation objective, sharing similar GRU architecture, as opposed to the more advanced DPRNN’s architecture and its end-to-end SE objective. The results signify the importance of the teacher model’s performance. As each fine-tuned student models stem from the same pretrained GRU model, this shows that the fine-tuned performance depends on the quality of the teacher model. It is also noticeable that the structural discrepancy between the student and teacher, i.e., $S_{\text{GRU}}$ (a GRU) and $T_{\text{DPRNN}}$ (a DPRNN), is not an issue. It implies that the proposed framework can potentially employ various advanced teacher models as the deep learning research improves the state of the art in the future.

We also observe that $S_{\text{DPRNN}}$ can catch up to the performance of $S_{\text{GT}}$, which is only marginally better. This suggests that fine-tuning on imperfect pseudo targets generated by $T_{\text{DPRNN}}$ has almost the same benefits as when ground-truth targets are used. Given the student models’ small architecture and limited generalization capacity, our personalization procedure can fine-tune the student model to its optimal performance but by relying only on the teacher’s SE results.
After personalization, the small student models consistently show significant improvements on their pretraining-based initialization. Hence, it verifies that our personalization framework is a model compression method if we compare the improved PSE models to those pretrained generalist models. Indeed, a smaller personalized model can compete with a large generalist, e.g., $2^{32}/C2^32$ DPRNN vs $2^{1024}/C2^{1024}$ S for $5$ dB input SNR as in Fig. 3(a). According to Table II, a personalized $2^{32}/C2^32$ specialist saves 11.99 M parameters and 756 M MACs compared to a $2^{1024}/C2^{1024}$ generalist (for 1-s inputs), which is more than 99% reduction in terms of spatial and arithmetic complexity. Therefore, even if further compression methods, e.g., eight-bit quantization, are always available to the $2^{1024}/C2^{1024}$ model, it will still most likely be more complex than $2^{32}/C2^{32}$ DPRNN. Furthermore, applying compression on larger models will subsequently lower their performance depending on the type and amount of compression. On the contrary, the $2^{32}/C2^{32}$ DPRNN outperforms the $2^{1024}/C2^{1024}$ $S$ even after its 99% reduction of complexity. This demonstrates that our framework works as a mode of lossless model compression. Thus, we argue that it is more advantageous to personalize the models instead of increasing generalists’ computational capacity for better generalization capabilities. In addition, as PSE shows improved performances in denoising and dereverberation tasks in various unique test environments, our personalization framework can be seen as a genuine adaptive system that specializes not only in each individual user but in the specific noise source and reverberant condition of the test time environment.

Figure 4 show the SE performance of various models on the reverberation-only input signals, i.e., with no additive background noise. It gives a separate view to the proposed PSE method’s dereverberation performance from the joint denoising and dereverberation setup. In addition, the dereverberation results are shown separately as two cases in which one-half of the environments ($K = 22$) are with low input scale-invariant SDR (SI-SDR) [Fig. 4(a)] and the other with high input SISDR [Fig. 4(b)]. For the lower SI-SDR cases, the results in Fig. 4(a) show that our framework can successfully personalize to different room acoustics. Contrary to joint dereverberation and denoising results, the improvements for $\sim S_{DPRNN}$ are minimal compared to those for $\sim S_{GRU}$. This trend is not observed in Fig. 4(b), which reports results from upper SI-SDR cases. First, the generalist models, S, worsen the sound quality. As the SI-SDR of the reverberant inputs were already high, the pretrained generalists must have injected artifacts that significantly decrease the quality of the signal. This is also evident in the $\sim S_{GT}$ baselines, where the model is fine-tuned on ground-truth anechoic targets. Hence, the goal of personalization in these cases is to be able to mitigate this performance degradation of the processed signals. Indeed, $\sim S_{DPRNN}$ show improved performances over the initial worsened estimates by the pretrained generalists, demonstrating the KD framework’s consistent capacity to produce pseudo-labels for fine-tuning student models.

**B. Subjective performance analyses on real-world test environments**

Additionally, we conducted a subjective listening test with ten participants using real-world noisy reverberant...
recordings from the voiceHome-2 dataset (Bertin et al., 2019). We trained ten personalized models from ten test speakers, whose noisy and reverberant samples were recorded in different rooms from different houses with varying speaker position and background noise types according to the setup of the voiceHome-2 dataset. Because the dataset was recorded in the real-world test environment, there is no clean utterance available for supervised learning, making our experiment realistic. The test was performed by asking ten listeners for their perceptual evaluation of the test sequences. Figure 5 presents their mean opinion scores (MOS). Each trial consists of an input noisy and reverberant sample, y, the enhancement result for a small student model, S, the output for the fine-tuned student model, \( \tilde{S}_{DPRNN} \), and the result for the DPRNN teacher model, T_{DPRNN}. The student models are with 64 hidden units. From Fig. 5, we can observe that the personalized student models \( \tilde{S}_{DPRNN} \) outperforms the baseline student models \( S_{DPRNN} \). The statistically significant improvement aligns with the objective metrics. Whereas this dataset contains only indoor recordings, this provides additional analysis not only regarding the significance of the metrics but also the effectiveness of the PSE framework.

C. Evaluation of personalization using varying amounts of fine-tuning datasets

Our proposed personalization showed great improvements under 5 min of fine-tuning data, which is noisy and reverberant speech recorded from the same acoustic scene during test time. However, we cannot assume this amount of data to be readily available for realistic scenarios. Hence, we test our framework on varying amounts of noisy reverberant mixtures as well, i.e., 10 s and 1 min. Figure 6 shows the average SI-SDR improvements by using varying lengths of noisy input data across all K environments. We only use the DPRNN teacher’s estimates for fine-tuning as we have observed its effectiveness from Sec. IV A. For brevity, we test on 0 and 10 dB input SNR cases.

As expected, the length of available datasets for fine-tuning is proportional to the test time performance of the personalized student models. This experiment illustrates a realistic use-case where, initially, 5 min of noisy data will not be directly available but rather 10 s and, later, 1 min of test time signals will be gradually collected over time in a realistic data collection scenario. From both figures in Fig. 6, we can observe that smaller personalized models can still outperform a larger generalist even with less fine-tuning data. For example, in Fig. 6(a), \( 2 \times 256 \tilde{S}_{DPRNN} \) personalized on only 10 s of data can outperform the largest generalist.

D. PSE models’ generalization performance on unseen speakers, noises, and room RIRs

Personalization could potentially worsen the generalization performance if a model fine-tuned on a specific test environment must generalize to other unseen test environments comprised of unseen speakers, noise types, or room conditions. The performance degradation is mainly due to the catastrophic forgetting phenomenon (French, 1999): fine-tuning on the target test time environment changes the weights that were initially pretrained on the general-purpose training set. This can be problematic if the model is relocated or the surrounding is changed (e.g., furniture rearrangement or new additions to room, such as draping, that could alter the acoustics).

We examine this behavior by challenging an already personalized student with a different unseen environment. For this experiment, we design \( K = 8 \) different environments with balanced speaker gender, noise class, and various room dimensions. Details of the configurations can be found in
Table IV. Further details on dimensions of the rooms can be found in Table VI in the Appendix. The student models are personalized to each th environment, using noisy reverberant signals generated from $s_{j,k}^{(i)}$, $h_{j,k}^{(i)}$, and $R_{j,k}^{(i)}$ for $K$ different personalized student models in total. The fine-tuned models are then evaluated on each th room using set-aside unseen datasets $s_{j,k}^{(i)}$, $h_{j,k}^{(i)}$, and $R_{j,k}^{(i)}$ taken from the same $K$ configurations, i.e., $j \in \{1, \ldots, K\}$. Thus, $k = j$ is the desired personalization setup, while $j \neq k$ represents the th PSE model challenged to work on the th environment. $2 \times 64$ RNN student and DPRNN teacher was used to produce Fig. 7. Additive background noise was scaled to 0 dB input SNR.

In Fig. 7, we show the relative differences between the pretrained generalist and personalized student models evaluated on all $K = 8$ environments. We report the result using SI-SDR, short-time objective intelligibility (STOI; Taal et al., 2011), and perceptual evaluation of speech quality (PESQ; Rix et al., 2001) scores for an in-depth evaluation on personalization and its following effects on other environments. We apply various metrics, SI-SDR, STOI, and PESQ, to provide a comprehensive measure of the noisy and reverberant conditions. It is not straightforward to find a metric for enhancement methods that necessarily leads to improvements for different downstream applications because different metrics capture different distortion measures. Considering automatic speech recognition (ASR) as an example, word error rate (WER) and STOI have shown a higher correlation coefficient than other objective evaluation metrics; however, under realistic conditions, there are various factors that affect the ASR performance, and there is no single metric that has been shown to necessarily lead to better WER (Chai et al., 2018; Fukumori et al., 2013).

The negative values in the cells indicate performance degradation incurred from personalization. Each th row corresponds to the performances of the model personalized on th environment and evaluated on the th environment. For example, the first row corresponds to the performances of the student model fine-tuned in environment “A” evaluated on all $K$ configurations.

Unsurprisingly, the diagonal axes generally show highest improvements because those cells represent evaluation results of student models personalized on the same environment. This supports the main argument of our proposed framework. On the other hand, there are underperforming cases, such as models fine-tuned on “E,” performing poorly on A (−1.1 dB ASI-SDR) and “F” (−0.9 dB ASI-SDR). Interestingly, the inverse relationship does not always hold. Although the model personalized on “B” generalize well to A (1.0 dB ASI-SDR), the model fine-tuned on A does not for B (−0.2 dB ASI-SDR). These results show that personalizing on one condition can incur negative effects when the model has to generalize to another environment.

Although this reveals a weakness of the proposed framework, this problem can be addressed with a simple solution by resetting or readjusting the model when sudden worsened performance is detected. However, it is not straightforward to detect such a performance drop during the test time. As a remedy, we propose to compare the PSE result to the teacher model’s estimate as an indirect way to evaluate the speech quality. It is because there are no ground-truth test time data available. In Fig. 8, we show the results from a same experimental setup with Fig. 7 but with the scores computed from using the teacher’s estimates as the reference, i.e., the pseudo targets. We see that the SI-SDR, STOI, and PESQ scores measured against teacher’s estimates (Fig. 8) are different from those measured against the ground-truth targets (Fig. 7). However, the scores measured using teacher’s outputs are still close approximates to the ground-truth metrics. This showcases another merit of using the teacher’s estimates. We can reliably use these pseudo metrics to estimate a model’s test time performance and decide to reset back to the pretrained generalist version or initiate a fine-tuning process to adjust the model to the new test environment.

Figures 7(b) and 7(c) also show that the relative differences in SI-SDR, STOI, and PESQ are not always correlated to one another. Cells with high relative SI-SDR improvements do not necessarily show improvements in intelligibility (e.g., model personalized on B evaluated on A). This could be a result of the loss function defined by $\Delta$ SI-SDR to optimize the student models during the fine-tuning process. A better optimization objective could be explored to prevent such differences. Nonetheless, personalization on intended environments generally shows large improvements without significant performance degradation.
E. Generalization performance to unseen locations within the same room

Next, we evaluate on a scenario in which a student model is personalized on a single location of a room and tested on unseen positions within the same room. Thus far, our experiments used all RIRs, \( R^{(k)}_i \), from the \( k \)th room to construct the test time fine-tuning dataset. It was to make the PSE model robust to the variations of RIR filters, which vary vastly depending on the microphone-speaker distance, vicinity to walls or corners, occluding objects, and other factors (Shinn-Cunningham et al., 2005). In this subsection, we fine-tune a student model using noisy reverberant data generated using a RIR signal from a specified location, \( i \), within the same \( k \)th room, \( R^{(k)}_i \). As with other experiments, utterances from a single user, \( S^{(k)}_i \), and one noise type, \( M^{(k)}_i \), are used for fine-tuning. For evaluation, we select RIRs from unseen speaker locations within the same room, \( R^{(k)}_j \), where \( j \neq i \). Unseen speech samples from \( S^{(k)}_t \) and noise sources, \( M^{(k)}_t \), are used to generate the noisy reverberant evaluation set. For brevity, we experiment on \( 2 \times 64 \) student models and 0 dB input SNR for additive background noise.

Two rooms from the BUT were selected for this experiment: a small office (L212) and a large conference room (D105). Their speaker locations and distances from the microphone are described in Table V. Geometric information of all the other rooms can be found in Table VI in the Appendix.

Figure 9(a) shows several speaker positions from room L212 used in this experiment. We fine-tune on noisy reverberant speech from an arbitrary position and evaluate on other locations of the room. We experiment using the same room but on multiple different speakers and noises described in Table IV. Figure 9(b) shows the average generalization results. Despite the changes in location of the test time speech source, fine-tuning on one location of a room can improve dereverberation and denoising results for unseen locations within the same room at least to some degree. The same behavior can be observed in Fig. 10 for a much larger room, D105, although the generalization power drops drastically when the unseen location is too different from that used for fine-tuning, e.g., as in \( F \). This demonstrates that a stationary personalized device is capable of performing robust SE on a nonstationary user within the same room as opposed to suffering from drastic changes in entire room geometry (Sec. IV D).
TABLE V. Locations and source-mic distance of the rooms.

<table>
<thead>
<tr>
<th>Room ID</th>
<th>Location</th>
<th>Location ID</th>
<th>Source-mic distance (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>L212</td>
<td>A</td>
<td>0.41 x 1.13 x 1.98</td>
<td>4.87</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>6.96 x 0.77 x 1.98</td>
<td>2.11</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>5.34 x 2.48 x 1.39</td>
<td>0.90</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>4.72 x 1.32 x 1.88</td>
<td>0.87</td>
</tr>
<tr>
<td></td>
<td>E</td>
<td>3.21 x 1.67 x 0.46</td>
<td>2.12</td>
</tr>
<tr>
<td>D105</td>
<td>A</td>
<td>11.93 x 22.93 x 3.63</td>
<td>13.76</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>5.23 x 9.90 x 1.82</td>
<td>4.18</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>14.79 x 20.79 x 4.47</td>
<td>14.76</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>6.01 x 6.06 x 1.97</td>
<td>7.85</td>
</tr>
<tr>
<td></td>
<td>E</td>
<td>9.65 x 6.22 x 3.12</td>
<td>10.01</td>
</tr>
<tr>
<td></td>
<td>F</td>
<td>0.70 x 5.48 x 2.02</td>
<td>7.86</td>
</tr>
</tbody>
</table>

V. CONCLUSION

In this paper, we proposed a zero-shot KD approach to personalizing SE models for joint dereverberation and denoising. Our goal was to adapt a small model to dynamically changing test time SE environment instead of employing a large generalist model, which can be too heavy for embedded systems. In doing so, we exploited widely available noisy mixtures during test time rather than leveraging ground-truth targets or any extra information of the acoustic environment, which are rarely available in the real-world use-cases. To improve the usability of the corrupt examples found in the test scene, our framework synthesizes pseudo targets by executing a superior-quality SE routine on an overly complex teacher model. We suggest that this KD-based personalization can be performed on a regular basis or when a significant change is detected in the test time acoustic scene. As this fine-tuning task can be performed either in the cloud or when the device is idle, we envision that it is not burdensome for the device.

Evaluation results demonstrate that the student model’s performance greatly improves on specific test time speakers and acoustic environments. The improvements were consistent under various noise and room conditions. Furthermore, the improvements can be observed regardless of model size or the amount of fine-tuning data available: the fine-tuned performance is dependent on the amount of data, but this does not pose a serious limitation on our framework as we can observe improvements even with minimal data. It is also noticeable that the architectural difference between the student and teacher models does not impact the personalization process. Therefore, we expect that our proposed framework can benefit from advancements in the future deep learning-based SE research. Because our small personalized student model can give superior performances to large generalist models, we claim that the KD-based fine-tuning method provides another mode of model compression that does not sacrifice performance, i.e., lossless model compression.

Although fine-tuning on specific environments can harm the generalization on other unseen scenes, the teacher’s estimates can again be used to gauge the change of environments. A decision can be made to reset the student’s parameter back to its pretrained value, followed by another personalization procedure for further adaptation. Also, our study shows that models personalized on one location can still show improved generalization on unseen locations within the same room, demonstrating robustness to nonstationary sources.

Major limitations of our current study for PSE comes from the dependency on the quality of the teacher model and especially the amount of fine-tuning data available. Another weakness is our experiments tested on test time environments contain only a single speaker and one unique noise source per room. Future research shall consider expanding this study to minimize the number of utterances required for the KD procedure and perform SE and separation under multi-speaker conditions.

ACKNOWLEDGMENTS

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

AUTHOR DECLARATIONS
Conflict of Interest

The authors have no conflicts to disclose.

DATA AVAILABILITY

The data that support the findings of this study are openly available in Librispeech (Panayotov et al., 2015),
APPENDIX: DIMENSIONS OF ROOMS

<table>
<thead>
<tr>
<th>Room ID</th>
<th>Size (m x m x m)</th>
<th>Volume (m³)</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q301</td>
<td>10.7 x 6.9 x 2.6</td>
<td>192</td>
<td>Office</td>
</tr>
<tr>
<td>L207</td>
<td>4.6 x 6.9 x 3.1</td>
<td>98</td>
<td>Office</td>
</tr>
<tr>
<td>L212</td>
<td>7.5 x 4.6 x 3.1</td>
<td>107</td>
<td>Office</td>
</tr>
<tr>
<td>L227</td>
<td>6.2 x 2.6 x 1.4</td>
<td>229</td>
<td>Stairs</td>
</tr>
<tr>
<td>R112</td>
<td>4.4 x 2.8 x 2.6</td>
<td>~400</td>
<td>Hotel room</td>
</tr>
<tr>
<td>CR2</td>
<td>28.2 x 11.1 x 3.3</td>
<td>1033</td>
<td>Conference room</td>
</tr>
<tr>
<td>E112</td>
<td>28.2 x 11.1 x 3.3</td>
<td>~900</td>
<td>Lecture room</td>
</tr>
<tr>
<td>D105</td>
<td>17.2 x 22.8 x 6.9</td>
<td>~2000</td>
<td>Lecture room</td>
</tr>
<tr>
<td>C236</td>
<td>7.0 x 4.1 x 3.6</td>
<td>102</td>
<td>Meeting room</td>
</tr>
</tbody>
</table>

MUSAN (Snyder et al., 2015), ESC50 (Piczak, 2015), AIR (Jeub et al., 2009), PORI (Merimaa et al., 2005), RWCP (Nakamura et al., 2000), BUT (Szöke et al., 2019), and REVERB (Kinoshita et al., 2013).

TABLE VI. Descriptions of rooms from BUT.

2See https://github.com/mpariente/pystoi (Last viewed December 2023).
3See https://github.com/vBaiCai/python-pesq (Last viewed February 16, 2022).

APPENDIX: DIMENSIONS OF ROOMS

TABLE VI describes the detailed geometric information about the BUT used in our experiments.


