RESOURCE-EFFICIENT MODEL ADAPTATION METHODS FOR PERSONALIZED SPEECH ENHANCEMENT SYSTEMS

Aswin Sivaraman

Submitted to the faculty of the Graduate School
in partial fulfillment of the requirements
for the degree
Doctor of Philosophy
in the Luddy School of Informatics, Computing, and Engineering,
Indiana University

May 2024

Accepted by the Graduate Faculty, Indiana University, in partial
fulfillment of the requirements for the degree of Doctor of Philosophy.

Doctoral Committee

Minje Kim, Ph.D., Chair

David Crandall, Ph.D.

Chris Raphael, Ph.D.

Ariful Azad, Ph.D.

03/22/2024

To my wife, my parents, my brother, my in-laws, and all my gurus over the years.

Aswin Sivaraman

RESOURCE-EFFICIENT MODEL ADAPTATION METHODS FOR PERSONALIZED SPEECH ENHANCEMENT SYSTEMS

This dissertation introduces several machine learning algorithms for developing personalized speech enhancement (PSE) systems. In particular, we investigate the data-efficiency of the proposed methods. Here, we define personalization as a model adapting towards a particular user's speech characteristics and/or their acoustic environment. By consciously minimizing their computational overhead, we make these algorithms more suitable for edge computing applications—e.g., smartphones, smart speakers, or headphones.

These use cases can all benefit from employing PSE systems on at least two dimensions. Firstly, PSE can lead to better performance—this is because single-user speech enhancement may be viewed as a subset of the originally complex problem (i.e., speaker-agnostic or general-purpose speech enhancement). Secondly, PSE can reduce model complexity; given the reduced problem space, a personalized model with fewer parameters suffices to perform equally as well as a non-personalized model trained with many more parameters. To that end, we argue that PSE is a novel paradigm for lossless model compression without loss of performance. However, PSE can be challenging from an optimization perspective. When framed as a fully supervised machine learning problem, the availability of labeled speaker-specific data is scarce, and attempting to collect user data may be unreliable and privacy-compromising.

To that end, this dissertation proposes data-efficient PSE methods that can tackle two potential scenarios. In the first case, the PSE system may have access to abundant unlabeled noisy speech data but only a small amount (up to 30 seconds) of clean speech data from the target user. In the second case, the PSE system may have no access to any personally identifiable data. Therefore, our methods may be classified as few-shot or zero-shot machine learning approaches.

In order to best utilize the scarce clean data in the few-shot context, we put forward self-supervised learning methods for PSE that repurpose the more accessible unlabeled speech data. More specifically, we develop frameworks that incorporate noisy target training and contrastive learning. Furthermore, to achieve zero-shot personalization, we employ the model selection paradigm for finding a predefined latent cluster best-suited for the unseen test time user's noisy speech.

Our extensive experiments show that both self-supervised learning and the model selection paradigm achieve our goals for model adaptation. This research promotes the development of more efficient speech enhancement systems with reduced training data requirements and broader accessibility for more people.

TABLE OF CONTENTS

ACCEPTANCE PAGE ii

DEDICATION iii

ABSTRACT iv

LIST OF ABBREVIATIONS ix

Chapter 1 Introduction 1

- 1.1 Problem Setup 3
- 1.2 Broader Impact 4
- 1.3 Relevant Publications 6

Chapter 2 Literature Review 7

- 2.1 Datasets & Benchmarks 7
- 2.2 Evaluation Metrics & Loss Functions 8
- 2.3 Training Targets 10
- 2.4 Resource Efficiency 11
- 2.5 Model Compression 13
- 2.6 Target Speaker Extraction 14
- 2.7 Self-Supervised Learning 15
- 2.8 Mixture of Local Experts 17
- 2.9 Non-Negative Matrix Factorization 19
- 2.10 Universal Speech Models 20

Chapter 3 Personalization through Noisy Target Training 22

- 3.1 Transfer Learning 23
- 3.2 Self-Supervised Feature Learning 25

3.2.1 Pseudo Speech Enhancement 27
3.2.2 Contrastive Mixtures 29
3.2.3 Data Purification 33
3.3 Experiment 37
3.3.1 Setup 37
3.3.2 Datasets 38
3.3.3 Neural Network Architectures 40
3.3.4 Implementation Details 42
3.3.5 Results 45
3.3.6 Summary 51
Chapter 4 Personalization through Test-Time Model Selection 53
4.1 Sparse Ensemble of Specialists 53
4.1.1 Specialist Networks 54
4.1.2 Gating Network 55
4.1.3 Ensemble Network 56
4.1.4 Experiment Setup 57
4.1.5 Results 59
4.1.6 Summary 61
4.2 Speaker-Informed Sparse Ensemble of Specialists 62
4.2.1 Design 63
4.2.2 Pretraining Process 64
4.2.3 Experiment Setup 68
4.2.4 Results 69
4.2.5 Summary 71
4.3 Block-Sparse RNNs for Universal Speech Modeling 73

4.3.1 Design . . . 74

4.3.2 Pretraining Process . . . 80

4.3.3 Experiment Setup . . . 81

4.3.4 Results . . . 85

Chapter 5 Conclusion . . . 88

5.1 Contributions . . . 90

5.2 Limitations & Future Work . . . 91

BIBLIOGRAPHY . . . 93

CURRICULUM VITAE . . .

LIST OF ABBREVIATIONS

ASR	Automatic Speech Recognition
BSGRU	Block-Sparse Gated Recurrent Unit
BSS	Blind Source Separation
CAE	Collaborative Audio Enhancement
CE	Cross-Entropy
CM	Contrastive Mixtures
CNN	Convolutional Neural Network
DAE	Denoising Autoencoder
DL	Deep Learning
DNN	Deep Neural Network
DP	Data Purification
EM	Expectation-Maximization
FFT	Fast Fourier Transform
FLOP	Floating-point Operation
FSL	Few-Shot Learning
GAN	Generative Adversarial Network
GMM	Gaussian Mixture Model
GRU	Gated Recurrent Unit
HMM	Hidden Markov Model
IBM	Ideal Binary Mask
ICA	Independent Component Analysis
IRM	Ideal Ratio Mask
LDA	Latent Dirichlet Allocation
LSH	Locality Sensitive Hashing
LSTM	Long Short-Term Memory
ML	Machine Learning
MAC	Multiply-Accumulate
MLE	Mixture of Local Experts
MSE	Mean-Squared Error
NMF	Non-Negative Matrix Factorization
NTT	Noisy-Target Training
PCA	Principal Component Analysis
PESQ	Perceptual Evaluation of Speech Quality
PLSI	Probabilistic Latent Semantic Indexing
PSE	Personalized Speech Enhancement
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
SAR	Signal-to-Artifact Ratio
SDR	Signal-to-Distortion Ratio
SE	Speech Enhancement
SIR	Signal-to-Interference Ratio
SISDR	Scale-Invariant Signal-to-Distortion Ratio
SI	Speaker Identification
SNR	Signal-to-Noise Ratio
SSL	Self-Supervised Learning
SS	Source Separation
STFT	Short-Time Fourier Transform
STOI	Short-Time Objective Intelligibility
SV	Speaker Verification
t-SNE	t-distributed Stochastic Neighbor Embedding
TF	Time-Frequency
TSE	Target Speaker Extraction
TL	Transfer Learning
TTS	Text-to-Speech
USM	Universal Speech Model
VAE	Variational Autoencoder
VCD	Voice Controlled Device
ZSL	Zero-Shot Learning

Chapter 1

Introduction

Every human voice is as unique as a fingerprint, filled with subtle nuances that comprise a portion of one's identity. While the physiological process of producing speech is the same for everyone, the resulting auditory signal contains numerous discerning features. A person's vocal range, their accent, and their speaking cadence can potentially be mapped to their locale, their age, or even their ethnic background. Because speech is such a deeply personal and invaluable biometric, it is quite unsurprising that humans have a complicated relationship with machine learning (ML)-based speech processing systems.

For the most part, people expect their devices to hear them in any noisy environment. Typically, voice controlled devices (VCDs) employ a general-purpose speech enhancement (SE) algorithm that improves the quality and intelligibility of the incoming speech signal. Currently, deep neural networks (DNNs) have become the de facto building blocks for modern SE algorithms. At the same time, it is well-known that generalizable DNN performance strongly correlates to increased model size and massive labeled datasets. In other words, developing a DNN for SE requires accruing vast amounts of training data (thousands of isolated speech or noise recordings) in order to cover the potential breadth of noisy speech signals that may be encountered at test-time [1]. Since every human voice is distinct, the SE model, by definition, will have never encountered the target speaker's voice during training, even though it is expected to be performant in this cold-start scenario. As a result, most SE models may be considered as "*generalists*" that assume no knowledge of the test-time environment. These generalists operate irrespective of the deployment context, intended to be universally applicable.

There are some notable downsides to models that targets large-scale generalized performance. For example, studies have shown that generalist DNNs possess redundant connections and under-utilized parameters [2]. With many models, performance and accuracy scales logarithmically with model capacity, potentially saturating after a point [3]. A lot of popular and emerging DNN architectures have started to exceed millions (or even billions) of model parameters, leading to skyrocketing hardware costs and exponentially high carbon footprints [4]. Furthermore, ML models trained on big data tend to exhibit sociodemographic biases [5, 6]—a phenomenon present in ML-based speech processing systems, too [7, 8]. Essentially, while generalist models may be well-performing, they incur a variety of resource inefficiencies along with the potential to fail for under-recognized people.

In an ideal setting, rather than attempting to generalize to every possible case, VCDs could instead utilize a personalized speech enhancement (PSE) model, adapted to enhance only the target speaker's voice optimally. Prior evidence has shown that the speech denoising problem can be decomposed into discrete non-overlapping sub-problems [9]. More specifically, the learning objective of a personalized specialist model (designed to enhance only a single voice) is simpler than that of a generalist model (which must enhance every possible voice), therefore a specialist may be better performing. One naïve method of adapting an SE model into a PSE model would be to fine-tune the model parameters using speaker-specific labeled data. Realistically, this data is often obtained through an "enrollment" procedure, where the target speaker records themselves saying a few prescribed sentences in a noise-free environment [10, 11, 12]. Once an SE model becomes specialized for a particular speaker or environment, the enhancement performance is expectedly improved, leading to a more robust on-device automatic speech recognition (ASR).

However, acquiring speaker-specific clean speech data is fraught with complications. Firstly, the average VCD user might not have access to very quiet echo-free environments or nice recording equipment; as a result, the user-provided data may not be considered to be reference quality. Secondly, typical ML models are trained on hundreds of hours of audio data, and recording any single speaker for that long would be unrealistic. At best, the burdensome enrollment procedure may yield a few seconds of usable data at most. Lastly, by and large, people are wary of AI-powered systems collecting too much of their personal data and breaching their privacy [13, 14, 15, 16]. It is understandable why people are reluctant to share their voice data given that vocal forgery is a legitimate concern. Recent research on speech synthesis models has shown that only 5 sec of enrollment data is needed to condition models into mimicking a particular voice [17]. As a result, minimizing the use of target speaker-specific data is a practical optimization constraint when developing a PSE model.

Therefore, the goal of this dissertation's proposed research is to reformulate ML algorithms for SE such that personalization can be achieved using little to none of the target speaker's personal data. Subsequently, we investigate how personalization not only improves performance for the target speaker but can also enable more efficient inference. More broadly, we posit that model adaptation (the idea of developing specialist models over generalist models) brings with it the added bonus of resource efficiency.

1.1 Problem Setup

The real-world deployment of a PSE model is subject to the aforementioned challenges of collecting target speaker specific data. Therefore, with this dissertation, we consider three possible scenarios pertaining to the availability of training data.

- **PSE Scenario 1 (Enrollment)**: The target speaker provides some amount of clean speech in order to optimize their experience. This set is commonly referred to as "enrollment data" and may be as little as 5 sec or at most 30 sec in total duration.

- **PSE Scenario 2 (Unlabeled)**: Here, the PSE system only has access to a few unlabeled observations of the target speaker. These "in-the-wild" recordings give the model some knowledge of the target speaker, but they are likely contaminated by unknown noises.
- **PSE Scenario 3 (Cold-Start)**: In this case, the target speaker provides no personal data of any kind. This is effectively a "cold start" problem.

We treat the first scenario as a few-shot learning (FSL) problem because it is about leveraging the scarcely labeled data without overfitting; accordingly, the last two scenarios may be treated as zero-shot learning (ZSL) problems due to the lack of enrollment data. For **PSE Scenario 2**, we hypothesize that the more abundant noisy data may be serviceable using a self-supervised learning (SSL) technique known as noisy-target training (NTT). Models pre-trained using the NTT methods can be fine-tuned over any available clean speech data, thereby covering **PSE Scenario 1**. To address **PSE Scenario 3** (or any instance of model adaptation without knowledge of the target domain), we propose the idea of "model adaptation by selection". Over all our experiments, we assess how the proposed algorithms achieve our adaptation goals of improving performance while enabling reductions in model complexity (either through quicker inference or fewer total parameters).

1.2 Broader Impact

This dissertation offers a preliminary exploration of two broad-impact areas of AI-based research. Our proposed methods to personalize an SE model meet the ever-growing need for ethical AI models that are more inclusive and responsible. The state-of-the-art machine learning models have gravitated towards those trained on the largest possible amount of data, often neglecting the representativeness of that data. Especially in SE research, the standard practice is either to record as many data samples as possible and then hire human annotators to label them, or to combine various publicly available datasets. Because data preparation and annotation are likely to incur the most significant costs when developing an AI system, researchers have overlooked their social impacts until recently. For example, one study showed that the accuracy of two ASR systems (on YouTube and Bing) was notably worse among non-American non-white female speakers [18]. Similar representation disparities, inherently caused by empirical risk minimization, have been surveyed in numerous machine learning tasks, including face recognition and language identification [19]. In most cases, the racial or gender inequities stemmed from the underlying biases in the large training datasets used. This dissertation addresses this ethics issue directly by developing speaker-specific specialist models that outperform speaker-agnostic generalist models. More broadly, we argue that specialist AI models can better serve socially under-represented groups.

In addition, our methods for PSE bring broader attention to the need for privacy-preserving AI systems. If a negligent party targets personalization solely as a means for increasing the accuracy of their AI systems, a breach of privacy is an imminent concern. The most apparent case is when an always-on VCD accidentally listens to a conversation due to mishearing the wake word [20]. Human employees might need to additionally annotate these utterances that caused the VCD to misfire, further diminishing the user's privacy. While privacy preservation has been investigated in other machine learning tasks (e.g., classification), it has been less studied with speech enhancement due to the need for clean speech from the test-time users. The experiments in this dissertation explicitly minimize the models' exposure to the target speakers' voices and their private environment. We hope that our investigations encourage further efforts by SE researchers to incorporate similar privacy constraints.

1.3 Relevant Publications

We acknowledge that certain portions of this dissertation consist of previously published material, presented at various conferences and journal articles. Table 1.1 summarizes the publication history of our prior works, indicating which sections use them. Additionally, we provide online access to source code and demos for the reader's reference.

Table 1.1: Archival links to the relevant prior publications, source code, and demos.

Year	**Publication Venue**	**Article Type**	**Citation**	**External Links**	**Relevant Sections**
2020	Interspeech	Conference	[21]	📄 🐙 🎥 🧪	Section 4.1
2020	NeurIPS	Workshop	[22]	📄 🐙 🎥	Section 3.2.2
2021	Interspeech	Conference	[23]	📄 🐙 🎥	Section 3.2.3
2021	WASPAA	Workshop	[24]	📄 🐙	Section 4.2
2022	JSTSP	Journal	[25]	📄 🐙	Chapter 3

Chapter 2

Literature Review

Although the human auditory system is exceptionally good at selective hearing [26, 27], no foolproof algorithm exists for perfectly emulating this ability computationally. In order to make machines capable of selective hearing, researchers either tackle the broader task of source separation (SS)—where the objective is to isolate individual sound sources from a mixture of sounds, or the narrower subproblem speech enhancement (SE)—where the objective is only to restore the speech source. Due to the ongoing deep learning renaissance, neural networks have surpassed traditional signal processing or machine learning methods, achieving state-of-the-art performance with both SS and SE tasks. In this chapter, we discuss the relevant prerequisite information for understanding the current lay of the land with SE research; additionally, we note the differences of the proposed methods from this dissertation with other influential works.

2.1 Datasets & Benchmarks

Currently, there is not one single benchmarking dataset for every SE model in existence, although there have been some public challenges attempting to bring standardization to the field [28, 29]. As a result, many researchers manually prepare noisy speech datasets (for training and for testing) by mixing utterances from public speech datasets[1] with sounds from public noise datasets[2]. Manually mixing utterances is a legitimate option as it enables fully-supervised training, with the caveat that researchers must apply a variety of signal-to-noise ratios (SNRs) in order to simulate varying degrees of noisiness. SNR

---

[1] Popular choices for English clean speech datasets include Librispeech [30] or Voice Bank [31].

[2] Popular choices for noise datasets include MUSAN [32] or DEMAND [33] or FSD50K [34].

is a measure of the ratio of speech power to noise power, often expressed in decibels; an SNR greater than 0 dB indicates the presence of more speech than noise. Some datasets are generated using a deterministic (fixed) mixing strategy to ensure reproducibility and standardization—for example, WSJ0-2Mix [35], LibriMix [36], and Voice Bank + DEMAND [37]. Our experiments in this dissertation make use of the open-source Librispeech, MUSAN, DEMAND, and LibriMix corpuses. Additional details about these datasets will be shared in the later sections.

2.2 Evaluation Metrics & Loss Functions

There are a number of evaluation metrics used when testing a SE model. Most metrics require access to the ground-truth (reference) clean speech, but a few are reference-less (i.e., blind quality estimators). Also, some metrics are considered signal level whereas others are perception level. In this dissertation, we will use metrics that do rely on the reference signal, i.e., in order to report objective improvements.

At the signal level, the most common metric is SNR—calculating the delta between the output (enhanced speech) SNR and input (noisy speech) SNR gives an indication of enhancement performance. Other prominent signal level metrics are modified versions of SNR, including signal-to-distortion ratio (SDR), signal-to-artifact ratio (SAR), and signal-to-interference ratio (SIR) [38]. Notably, SDR becomes equivalent to SNR when we only consider additive noise, ignoring interferences and algorithmic artifacts. More recently, a more robust modification of SDR known as scale-invariant signal-to-distortion ratio (SISDR) was proposed; it introduces a scaling factor to ensure that the residual vector—between the estimated and reference signals—maintains orthogonality to the reference [39]. In this dissertation, we will notate the reference (clean speech) signal as $\boldsymbol{s}$ and the model estimate (enhanced speech) signal as $\boldsymbol{y}$; the subscript t denotes the indexing of time-domain samples.

Subsequently, the metrics SDR and SISDR are computed in decibels as follows:

$$\mathrm{SDR}\left(\boldsymbol{y},\boldsymbol{s}\right) = 10\log_{10}\left[\frac{\sum_t {s_t}^2}{\sum_t (s_t - y_t)^2}\right] \tag{2.1}$$

$$\mathrm{SISDR}\left(\boldsymbol{y},\boldsymbol{s}\right) = 10\log_{10}\left[\frac{\sum_t (\alpha s_t)^2}{\sum_t (\alpha s_t - y_t)^2}\right] \tag{2.2}$$

Note that Eq. (2.1) is equivalent to setting $\alpha = 1$ in Eq. (2.2); however, as shown in [39], the robustness of SISDR comes from setting $\alpha = \left(\boldsymbol{y}^\top \boldsymbol{s}\right) / \left(\boldsymbol{s}^\top \boldsymbol{s}\right)$. All of these signal level metrics can, in fact, be used as optimization criteria for updating model parameters. When formulated as neural network loss functions (optimized for minimizing error), using the negative metric suffices [40]—in other words, a SISDR-based loss function would look like:

$$\mathcal{L}_{\mathrm{SISDR}}\left(\boldsymbol{y},\boldsymbol{s}\right) = -\,\mathrm{SISDR}\left(\boldsymbol{y},\boldsymbol{s}\right) = -10\log_{10}\left[\frac{\sum_t (\alpha s_t)^2}{\sum_t (\alpha s_t - y_t)^2}\right] \tag{2.3}$$

On the perceptual level, the metric STOI [41] (short-time objective intelligibility) measures speech intelligibility by calculating correlations between short-term temporal envelopes of the reference signal and of the enhanced signal; STOI values range between 0 and 1, where 1 would be most intelligible. Another metric, PESQ [42] (perceptual evaluation of speech quality), was introduced by the International Telecommunication Union (ITU)—PESQ also requires the reference signal, generating a score between -0.5 and 4.5 corresponding to a predicted perceptual MOS (mean opinion score). Both of these perceptual metrics can again be used as evaluation criteria but also as optimization criteria—i.e., as a neural network training loss function [43].

Lastly, while it does not necessarily guarantee high speech quality or intelligibility, another straightforward choice for neural network loss function is mean-squared error (MSE).

$$\mathcal{L}_{\text{MSE}}\left(\boldsymbol{y}, \boldsymbol{s}\right) = \sum_{t} \left(s_t - y_t\right)^2 \tag{2.4}$$

The models in this dissertation employ some of these discussed loss functions, namely $\mathcal{L}_{\text{MSE}}$ and $\mathcal{L}_{\text{SISDR}}$. In the later sections, where we introduce a classification sub-module, the conventional choice of loss function is the averaged cross entropy (log loss). For binary classification over N observations, given an array of ground-truth class labels $\boldsymbol{k}$ and an array of model-estimated labels $\hat{\boldsymbol{k}}$, the cross entropy (CE) optimization criterion is defined as:

$$\mathcal{L}_{\text{CE}}\left(\hat{\boldsymbol{k}}, \boldsymbol{k}\right) = -\frac{1}{N} \sum_{j=1}^{N} \left[k_j \log(\hat{k}_j) + (1 - k_j) \log(1 - \hat{k}_j)\right] \tag{2.5}$$

2.3 Training Targets

Most deep learning SE models proposed over the years can be broadly categorized in terms of their training targets, either as a masking-based or a mapping-based model [44]. Masking-based SE models operate on a two-dimensional time-frequency (TF) representation of audio; they learn to predict a binary masking matrix by processing the magnitude spectrum of a noisy speech signal [45, 46, 47, 48]. The magnitude spectrum is commonly obtained using the short-time Fourier transform (STFT). The masking matrix accentuates TF bins dominated by speech and filters out TF bins dominated by noise. In contrast, mapping-based models [49, 50] directly estimate a one-dimensional signal, the clean speech waveform. Fig. 2.1 shows a high-level comparison of the two training targets. For mapping-based models, $\boldsymbol{x}$ is the input mixture signal and $\boldsymbol{y}$ is the output estimated clean speech. If a time-frequency transform is used (e.g., the STFT), then the model input is the noisy speech magnitude

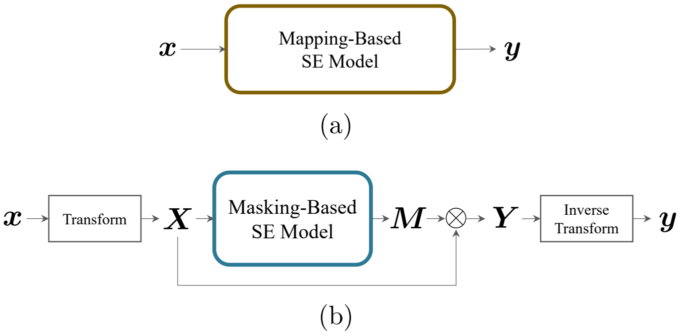


Figure 2.1: Comparison between mapping-based and masking-based SE models.

spectrogram $\boldsymbol{X}$ and the model output is a binary masking matrix $\boldsymbol{M}$. The operator $\otimes$ denotes element-wise multiplication (also known as the Hadamard product). The clean speech spectrogram is estimated by applying the mask, i.e., $\boldsymbol{Y} = \boldsymbol{X} \otimes \boldsymbol{M}$. An inverse transform is needed to convert time-frequency spectrograms back to time-domain waveforms. In a fully-supervised learning setup, the model's final estimate $\boldsymbol{y}$ is compared against the ground-truth clean speech $\boldsymbol{s}$. With our experiments in this dissertation, we employ both masking-based and mapping-based models, emphasizing that our proposed methods are agnostic to the choice of training target.

2.4 Resource Efficiency

The highly performant state-of-the-art models for SE are, in fact, double-edged. Because of the data-hungry nature of fully-supervised deep learning, many models for SE are likely to be over-parameterized, making them cumbersome both for training and for deployment on real-world devices. As stated before, a specialist model may be more resource-efficient compared to a generalist model given that it is solving a smaller sub-problem. In this dissertation, we address "resource efficiency" from multiple angles. For instance, we hypothesize that a personalized model may achieve equivalent performance to a generalist model using fewer model parameters. In that regard, personalization may be seen as a form of *lossless model compression*. By using fewer model parameters, we say that the PSE model has reduced

space complexity, i.e., its storage requirements are lessened. Whenever it is not possible to reduce the space complexity, we demonstrate that some personalized models can run fewer computations during inference compared to generalists. In other words, personalization can also reduce **run-time complexity**, improving latency or model throughput. Lastly, because a personalized model need only be optimized for one speaker (as opposed to thousands of speakers), we hypothesize that **training data reduction** is also possible. Storing massive datasets, synthesizing noisy speech, and updating the model parameters based on hundreds of hours of audio data is a very costly process. Particularly in Chapter 3, our proposed noisy-target training for personalization enables a specialist to use only 25 min of data in contrast to a generalist which uses 440 h of data, resulting in a massive 99.9 % savings. All of these benefits make PSE models more suitable for real-world deployment.

In relation to the previously mentioned public datasets, some of the reported best-performing deep learning models for generalist SE include ConvTasNet [49], dual-path RNN [51], SuDoRMRF [52], SepFormer [53], SCP-GAN [54], and MFNet [55]. We note that overall model complexity can be profiled using two measurements: for example, the total number of model parameters relates to space complexity, whereas the total number of multiply-accumulate operations (MACs) is indicative of run-time complexity. In Table 2.1, we list the number of total parameters and MACs for some top-performing models. Note that the number of MACs relates to the size of the model input—assuming that all audio recordings have a sampling rate of 16 kHz, we report the number of MACs for processing a single second of audio.

These state-of-the-art models achieve noteworthy improvements on enhanced speech SISDR[3], yet their space and run-time complexity are on the order of millions and billions, respectively. In particular, SepFormer is a massive neural architecture due to its use of transformer

---

[3] A leaderboard for speech separation performance on the WSJ0-2mix dataset [35] can be found at `https://paperswithcode.com/sota/speech-separation-on-wsj0-2mix`.

Table 2.1: Space and run-time complexities for various state-of-the-art SE models.

Model Name	**(# MACs)/sec.**	**# Params.**	**SISDR Imp.** [↑]
ConvTasNet [49]	9.82 G	4.92 M	15.3 dB
DPRNN [51]	15.24 G	3.63 M	18.8 dB
SuDoRMRF [52]	4.16 G	2.45 M	19.5 dB
SepFormer [53]	77.33 G	17.25 M	22.3 dB

layers [56]. As we discussed before, larger models require more expensive GPU hardware and incur a greater carbon footprint [57]. The investigations of this dissertation apply to models that operate below 1 M parameters, which are much more amenable to low-resource environments and real-world embedded systems. For that reason, the performance of the models discussed in this dissertation do not compete with and are not directly comparable to the state-of-the-art results shown in Table 2.1. Specifically in Chapter 3, we introduce and evaluate much smaller variants of ConvTasNet. Later on, we discuss how model adaptation (personalization) allows for equivalent performance to be achieved using smaller models.

2.5 Model Compression

Although model compression is an active area in deep learning research, many standardized methods, such as quantization or pruning [58], 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 [3]. Particularly with regards to SE or SS, more recent research has focused on novel model compression methods, including bitwise operations [59, 60, 61, 62] or group communication in intermediate neural network layers [63]. 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. This dissertation introduces a novel paradigm for lossless compression by means of personalization. For example, in Chapter 4, we demonstrate gender-based speaker adaptation via model selection, and our experiment results show that a specialist composed with 512 hidden units enhanced the target speaker's voice comparably to a generalist model composed with 1024 hidden units. This is effectively a "lossless" 50 % reduction in run-time computational complexity.

2.6 Target Speaker Extraction

Within the last few years, more and more SE research is being done on personalization, i.e., single-speaker model adaptation. However, the primary goal of most researchers is to show improved performance on the target speaker; the additional benefits of model compression, data efficiency, and privacy preservation are less explored. In the other literature, the PSE task is framed as target speaker extraction (TSE): effectively a combination of source separation (SS) with a conditional noise suppression (SE). This perspective of PSE is viable when there is some data from the target speaker available, i.e., enrollment data. Multiple models—such as SpeakerBeam [64], VoiceFilter [65], and pDCCRN [66]—explicitly utilize an encoder module that produces a noise-robust discriminative speaker embedding. As shown in Fig. 2.2, after the mixture sources are separated, the embedding cues the model to enhance only the desired source, i.e., the target speaker. Recent iterations of the public Deep Noise Suppression challenge [29] have included a 'personalized speech enhancement' track providing enrollment data, which effectively encourages participants to devise TSE solutions.

In contrast to these approaches which rely on computing a discriminative speaker embedding, our methods for PSE do the adaptation implicitly. That is, we do not uniquely ID the target speaker by way of a personally identifiable embedding—we intentionally do this

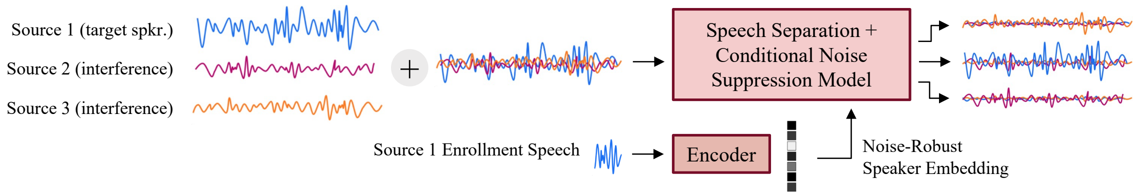

Source 1 (target spkr.)
Source 2 (interference)
Source 3 (interference)
+
Speech Separation + Conditional Noise Suppression Model
Source 1 Enrollment Speech
Encoder
Noise-Robust Speaker Embedding

Figure 2.2: Formulation of PSE as target speaker extraction (TSE).

to preserve the target speaker's privacy. Rather, we show that personalized enhancement can be achieved in a data-driven manner by using self-supervised learning (SSL) over the more-abundant in-the-wild observations of the target speaker, described in Chapter 3. Moreover, if no observations of the target speaker are available, we show that it is also possible to perform a coarse clustering on the test-time data, in order to enhance the target speaker's voice as it relates to a predefined group; this process of model adaptation by selection is described in Chapter 4.

2.7 Self-Supervised Learning

In the self-supervised learning (SSL) paradigm, a ML model is trained to solve a pretext task, learning useful features that will help when addressing the intended downstream task. The purpose is to overcome scenarios where there is only unlabeled data available (i.e., no input + expected output pairs). SSL has gained significant traction in recent years for advancing the state-of-the-art over numerous research domains, including speech representation learning [67, 68, 69]. There have also been a growing number of SSL setups 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 [70]. Another work describes an SSL framework based on two autoencoders, trained to reproduce either clean speech or noisy speech [71]. The authors enforce a coupling of the two autoencoders' latent spaces using cyclic-consistency. 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.

Subpar performance of an SE model can sometimes be attributed to a fundamental mismatch between the distribution of audio data encountered at training versus at test time. Because observations at test-time are inherently unlabeled, SSL is a great choice for enabling *in-domain training*. With the SE task, the term “unlabeled data” refers to the non-reference quality noisy speech recordings—which is likely more abundant. Two recent studies investigated using noisy speech data as target signals specifically to achieve in-domain training [72, 73]; because of the imperfect reference signals, noisy-target training (NTT) may be considered as an SSL pretext task. Our proposed PseudoSE method, introduced in Chapter 3, is also a form of NTT; however, this dissertation 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 NTT using contrastive learning for the task of SE.

There is also a well-regarded SSL framework for source separation (SS) known as mixture invariant training (MixIT) [74]. It was proposed as an alternative to the fully-supervised permutation invariant training (PIT). MixIT is a procedure for developing source separation systems using only mixtures of mixtures (MoM), i.e., linear combinations of arbitrary audio signals. When we consider MixIT as a pretext task, it introduces systematic mismatch by design because the input MoMs have twice the number of expected sources at test-time. One recent study used MixIT by successfully adapting models to a set of speakers through joint training over in-domain and out-of-domain data [75], 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. We discuss this further in Section 3.2.1.

2.8 Mixture of Local Experts

The mixture of local experts (MLE) modeling paradigm [76] has seen a few investigations in SE [77, 78], demonstrating that an ensemble of weak learners can produce a superior enhancement through a weighted combination of the learners' outputs. This general-purpose ensemble model is made up of two main components. First, multiple "expert modules" each learn to handle a subset of the complete set of training cases. Second, a classifier, referred to as the "gating module", is trained to predict a decision vector (p_k) that estimates the contribution of each expert with respect to the final output. As shown in Figure 2.3, the naïve output of an MLE ensemble model is simply the sum of the experts' individual inferences weighted by p_k. All expert modules receive the same input signal $\boldsymbol{x}$ and calculate their own expected outputs $\hat{\boldsymbol{s}}$. The gating module processes the input signal and outputs a normalized decision vector p_k that is used to combine the experts' outputs.

Instead of linearly combining the outputs of the separate experts, we imagine that the gating network makes a stochastic decision about which single expert to use on each occasion. More broadly, we propose swapping out the typical convex combination of the ensemble model to instead do model selection. By introducing "sparseness" in the output layer of the gating module, the MLE becomes more selective, effectively making a hard decision [21, 79]. Compared to generalist models, which require a large model capacity to achieve a certain level of speech denoising, a sparse ensemble model can yield the same enhancement quality even if the composing specialists use much fewer parameters. Subsequently, we claim that

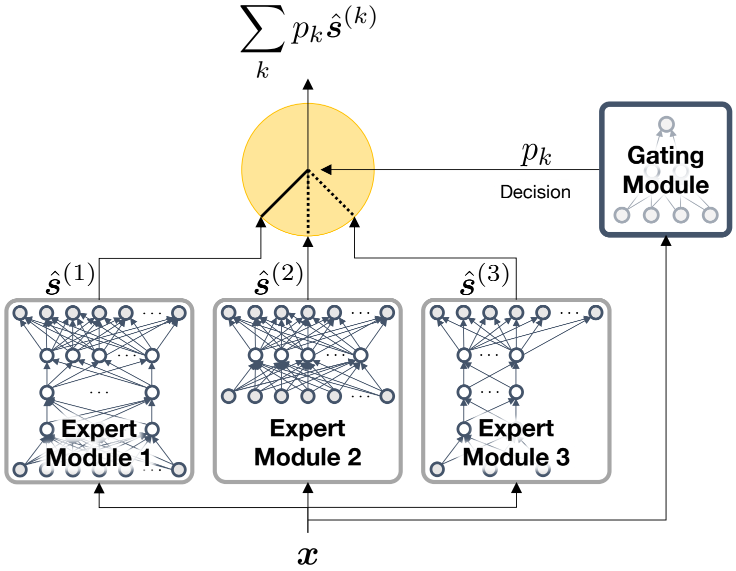

$\sum_k p_k \hat{\boldsymbol{s}}^{(k)}$
p_k
Decision
Gating Module
$\hat{\boldsymbol{s}}^{(1)}$
$\hat{\boldsymbol{s}}^{(2)}$
$\hat{\boldsymbol{s}}^{(3)}$
Expert Module 1
Expert Module 2
Expert Module 3
$\boldsymbol{x}$

Figure 2.3: An ensemble model based on the "mixture of local experts" paradigm.

our "sparse MLE" framework can also be a form of model compression. We provide specific implementation details of the sparse MLE model in Chapter 4.

2.9 Non-Negative Matrix Factorization

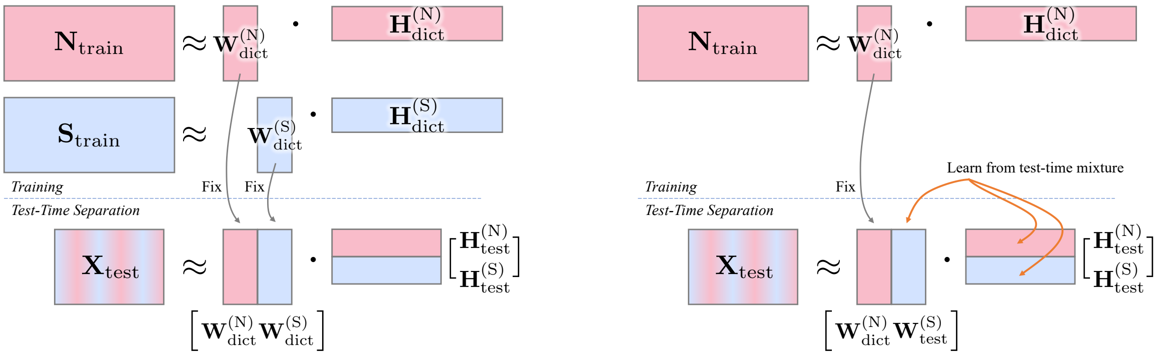
Prior to the advent of deep learning, earlier studies about adaptation through model selection looked at dictionary-based machine learning methods, such as non-negative matrix factorization (NMF) or probabilistic latent semantic indexing (PLSI) [80, 81, 82]. For example, when using NMF for speech enhancement, one common approach is to learn speech and noise spectrogram "dictionaries" (i.e., a set of basis vectors). To start, NMF solves the optimization problem:

$$\min_{W,H\geq 0} D(V||WH) \tag{2.6}$$

where D is a divergence function, V is a magnitude spectrogram, and W, H are learned factors. Because of the non-negativity constraint, W can be interpreted as the latent spectral features and H is their activation in time. Subsequently, the NMF pipeline for fully-supervised speech enhancement is as follows:

1. We first factorize the magnitude spectrogram of the training data speech corpus $\mathbf{S}_{\text{train}}$ and of the training noise corpus $\mathbf{N}_{\text{train}}$.
2. The resulting speech and noise basis vectors ($\mathbf{W}_{\text{dict}}^{(\text{S})}$ and $\mathbf{W}_{\text{dict}}^{(\text{N})}$) are kept fixed, treated as "dictionaries".
3. Next, the magnitude spectra of the mixture test signal $\mathbf{X}_{\text{test}}$ can be decomposed using the fixed dictionaries. The resulting test-time activation matrix ($\mathbf{H}_{\text{test}}$) can be partitioned similar to the dictionaries (into $\mathbf{H}_{\text{test}}^{(\text{S})}$ and $\mathbf{H}_{\text{test}}^{(\text{N})}$).
4. Finally, the clean speech estimate may be obtained by multiplying the factored matrices, i.e., $\mathbf{W}_{\text{dict}}^{(\text{S})}\mathbf{H}_{\text{test}}^{(\text{S})}$.

The success of this dictionary-based fully-supervised pipeline is bounded by the mismatch of the test-time data with the training data speech and noise templates. With a semi-supervised


N_train ≈ W^(N)_dict · H^(N)_dict
S_train ≈ W^(S)_dict · H^(S)_dict
Training
Fix
Fix
Test-Time Separation
X_test ≈ [W^(N)_dict W^(S)_dict] · [H^(N)_test H^(S)_test]
N_train ≈ W^(N)_dict · H^(N)_dict
Learn from test-time mixture
Training
Fix
Test-Time Separation
X_test ≈ [W^(N)_dict W^(S)_test] · [H^(N)_test H^(S)_test]

(a) (b)

Figure 2.4: Visual example of how non-negative matrix factorization (NMF) may be used for either (a) fully-supervised or (b) semi-supervised speech enhancement.

pipeline, we can relax this constraint; that is, as long as the test-time noise source is known, the test-time speech source can be learned. Both NMF approaches are illustrated in Fig. 2.4.

2.10 Universal Speech Models

universal speech models (USMs) extend the insights of the NMF semi-supervised pipeline. If the unknown source is surely a speech signal, then it may be approximated using a USM [83] (i.e., a set of templates for many different speakers). In the training stage for the USM, a speech dictionary is obtained by concatenating submatrices W_i which are the basis vectors of a training set speaker $i = 1, \ldots, M$, each obtained through a separate NMF decomposition, i.e., $\mathbf{W}^{(\mathrm{S})}_{\mathrm{dict}} = [\ W_1\ \ \ldots\ \ W_M\ ]$. If a noise model $\mathbf{W}^{(\mathrm{N})}_{\mathrm{dict}}$ is also available, then the speech enhancement task is simply a matter of estimating $\mathbf{H}_{\mathrm{test}}$. Because the USM is a larger model, which surely over-parameterizes the unknown test-time speaker, a block sparsity constraint is applied. This is reflected in a new optimization criteria:

$$\min_{W,H\geq 0} D(V||WH) + \lambda\, \Omega(H^{(\mathrm{S})}) \tag{2.7}$$

Based on the block sparsity function Ω, and with a sufficiently large choice for λ, Eq. (2.7) is a regularized version of Eq. (2.6) that encourages only a single speaker model to be active.

In other words, the USM iteratively converges on the best-fit training speaker. The authors show empirically that the USM (a speaker-independent model) achieves comparable results to fully-supervised NMF (a speaker-dependent model). Their block-sparse selection of the best-fit speaker is evidence to the claim that adaptation through model selection enables reduction in computational complexity. This is because the non-relevant basis vectors from other speakers are zeroed out using λ. In Chapter 4, we extend the idea of USM using a real-time deep learning framework which we call "block-sparse gated recurrent units" (BSGRUs). The proposed BSGRU has its learnable parameters subdivided into a variable number of learned groups, enabling frame-by-frame adaptation over time-varying audio signal characteristics, in place of speaker dictionaries.

Chapter 3

Personalization through Noisy Target Training

In this chapter, we consider few-shot methods for personalizing a speech enhancement system. As discussed in Chapter 1, achieving personalization without compromising the target speaker's privacy is of the highest priority. However, there may be cases wherein the target speaker consents to providing some small amount of personal data in order to facilitate an optimized experience. For example, some voice controlled devices (VCDs) have a one-time enrollment phase during setup, prompting the target speaker to recite a few template sentences. This process can be burdensome because if the "enrollment data" is not sufficiently intelligible, the device might need the speaker to re-record. Additionally, the service provider becomes responsible for storing the speaker-specific recordings securely on-device. Ultimately, the enrollment step may only yield a few seconds of total clean speech data from the target speaker. While this data is in-domain and useful for model adaptation, it is exceedingly few in comparison to standard datasets for training speech processing machine learning models, often containing thousands of hours of data. Therefore, the few-shot learning (FSL) problem for PSE is an optimization task of how to best utilize this scarcely available data without the possibility of model overfitting. In Section 2.6, we discussed target speaker extraction (TSE) as a popular approach for leveraging enrollment data. With this dissertation, we envision the worst-case and best-case amounts of available enrollment data to be either 5 or 30 s so as to minimize the concern of vocal forgery.

3.1 Transfer Learning

For our discussion, we assume a hypothetical set $\mathbb{T}$ that encompasses all of the target speaker's clean utterances. Given the privacy concerns and technical difficulties, we assume that this set is inaccessible to the training algorithm; therefore, it cannot be used for personalization. In **PSE Scenario 1**, the short recordings provided by the target speaker represent a small subset of their unavailable ground-truth clean speech, i.e., $\mathbb{T}_{\mathbf{f}\text{-tr}} \ll \mathbb{T}$. 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 $\mathbb{T}_{\mathbf{f}\text{-tr}}$. By doing this transfer learning, we adapt a generalist model into a specialist model.

Training a generalist requires a large set of many anonymous speakers $\mathbb{S}$ 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 $\boldsymbol{s} \in \mathbb{S}_{\text{tr}}$ 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 (SNRs). The generalist model can be described as a mapping function $\mathcal{G}$ with parameters $\mathcal{W}_{\text{SE}}$ which is trained such that $\mathcal{G}(\boldsymbol{x}; \mathcal{W}_{\text{SE}}) = \boldsymbol{y} \approx \boldsymbol{s}$, 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: $\mathcal{E}(\boldsymbol{y} \parallel \boldsymbol{s})$.

$$\mathcal{L}_{\text{SE}} = \mathcal{E}(\boldsymbol{y} \parallel \boldsymbol{s}) \tag{3.1}$$

$$\mathcal{W}_{\text{SE}} \leftarrow \operatorname*{arg\,min}_{\mathcal{W}_{\text{SE}}} \mathcal{L}_{\text{SE}} \tag{3.2}$$

There are many possible choices for the signal discrepancy function $\mathcal{E}$. The well-known signal-to-distortion ratio (SDR) metric [38] is frequently used as a general-purpose loss function for fully-supervised monaural time-domain speech enhancement [84]. 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:

$$\mathcal{E}_{\text{SDR}}(\hat{\boldsymbol{v}} \parallel \boldsymbol{v}) = -10 \log_{10} \left[\frac{\sum_t (v_t)^2}{\sum_t (v_t - \hat{v}_t)^2} \right]. \tag{3.3}$$

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.

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 $\mathbb{S}$ and the small target speaker-provided clean dataset $\mathbb{T}_{\textbf{f}\text{-tr}}$. To do this, we create speaker-specific artificial mixture signals $\boldsymbol{x}$ composed stochastically by sampling from the limited subset $\boldsymbol{s} \in \mathbb{T}_{\textbf{f}\text{-tr}}$ 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. (3.2), 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

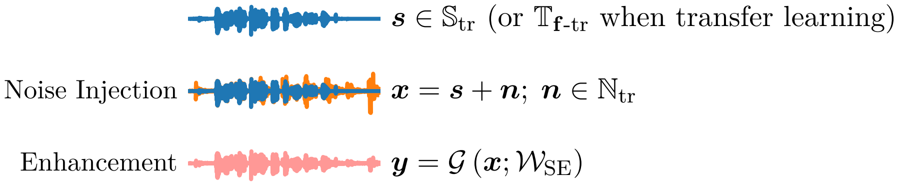

$\boldsymbol{s} \in \mathbb{S}_{\text{tr}}$ (or $\mathbb{T}_{\textbf{f}\text{-tr}}$ when transfer learning)
Noise Injection
$\boldsymbol{x} = \boldsymbol{s} + \boldsymbol{n};\ \boldsymbol{n} \in \mathbb{N}_{\text{tr}}$
Enhancement
$\boldsymbol{y} = \mathcal{G}\left(\boldsymbol{x}; \mathcal{W}_{\text{SE}}\right)$

Figure 3.1: Multi-speaker (fully-supervised) speech enhancement setup.

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 $\mathbb{T}_{\textbf{f}\text{-tr}}$. Suppose the number of signals within $\mathbb{T}_{\textbf{f}\text{-tr}}$ is too few; in that case, finetuning may fail to improve performance even though $\mathbb{T}_{\textbf{f}\text{-tr}}$ 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 $\mathbb{T}_{\textbf{f}\text{-tr}}$.

Fig. 3.1 shows a visualization of the baseline pretraining process. 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, the finetuning process is exactly the same as illustrated in Fig. 3.1; that is, $\boldsymbol{s}$ is sampled from the small speaker-specific dataset $\mathbb{T}_{\textbf{f}\text{-tr}}$, i.e., the enrollment data. The same signal transformations occur during transfer learning, when adapting the generalist model into a specialist model. If the target speaker does not provide $\mathbb{T}_{\textbf{f}\text{-tr}}$, the generalist model remains unadapted and therefore non-personalized.

3.2 Self-Supervised Feature Learning

Here we describe our proposed 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).

Our utilization of SSL stems from the assumption that *noisy* utterances from the target speaker $\tilde{\boldsymbol{s}} \in \tilde{\mathbb{T}}_{\mathbf{p}\text{-tr}}$ are much more available than clean ones, i.e., $|\tilde{\mathbb{T}}_{\mathbf{p}\text{-tr}}| \gg |\mathbb{T}_{\mathbf{f}\text{-tr}}|$. 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., $|\tilde{\mathbb{T}}_{\mathbf{p}\text{-tr}}| \gg |\mathbb{T}_{\mathbf{f}\text{-tr}}|$. Although we want to exploit these in-the-wild recordings $|\tilde{\mathbb{T}}_{\mathbf{p}\text{-tr}}|$, we do not know whether the observations are clean or noisy, i.e., the data is unlabeled. Therefore, we have to assume that $|\tilde{\mathbb{T}}_{\mathbf{p}\text{-tr}}|$ holds contaminated versions of some unobserved target clean speech signal $|\mathbb{T}_{\mathbf{p}\text{-tr}}|$. 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. 3.2 summarizes the training procedure of the baseline generalist-based pretraining, comparing it to our proposed SSL-based pretraining. 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

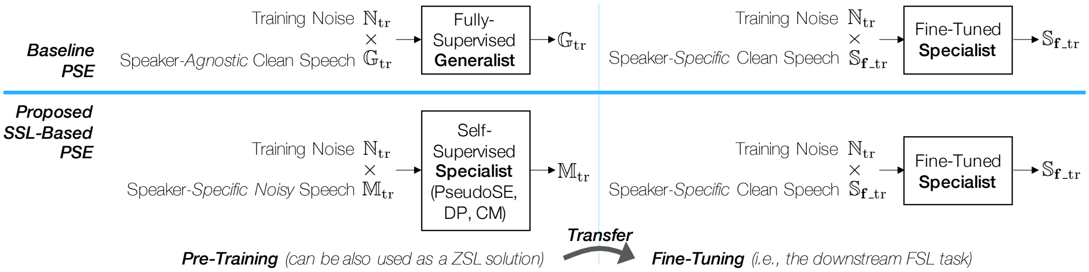

Baseline PSE
Training Noise $\mathbb{N}_{\text{tr}}$
×
Speaker-*Agnostic* Clean Speech $\mathbb{G}_{\text{tr}}$
Fully-Supervised **Generalist**
$\mathbb{G}_{\text{tr}}$
Training Noise $\mathbb{N}_{\text{tr}}$
×
Speaker-*Specific* Clean Speech $\mathbb{S}_{\mathbf{f}\text{\_tr}}$
Fine-Tuned **Specialist**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

Figure 3.2: An overview of the baseline and proposed personalization methods.

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. 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.

3.2.1 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 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 $\tilde{\boldsymbol{s}} \in \tilde{\mathbb{T}}_{\mathbf{p}\text{-tr}}$ 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 $\tilde{\boldsymbol{m}}$.

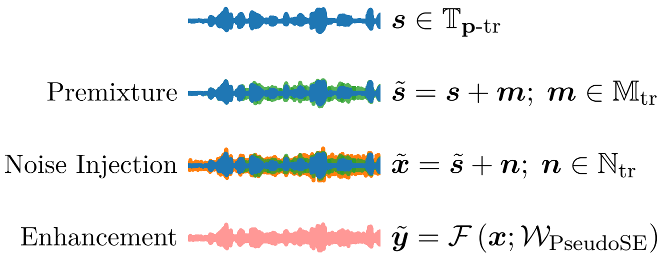

$\boldsymbol{s} \in \mathbb{T}_{\mathbf{p}\text{-tr}}$
Premixture
$\tilde{\boldsymbol{s}} = \boldsymbol{s} + \boldsymbol{m};\ \boldsymbol{m} \in \mathbb{M}_{\text{tr}}$
Noise Injection
$\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{s}} + \boldsymbol{n};\ \boldsymbol{n} \in \mathbb{N}_{\text{tr}}$
Enhancement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

Figure 3.3: Single-speaker (self-supervised) pseudo speech enhancement setup.

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

$$\mathcal{L}_{\text{PseudoSE}} = \mathcal{E}(\tilde{\boldsymbol{y}} \parallel \tilde{\boldsymbol{s}}) \tag{3.4}$$

$$\mathcal{W}_{\text{PseudoSE}} \leftarrow \underset{\mathcal{W}_{\text{PseudoSE}}}{\arg\min}\ \mathcal{L}_{\text{PseudoSE}} \tag{3.5}$$

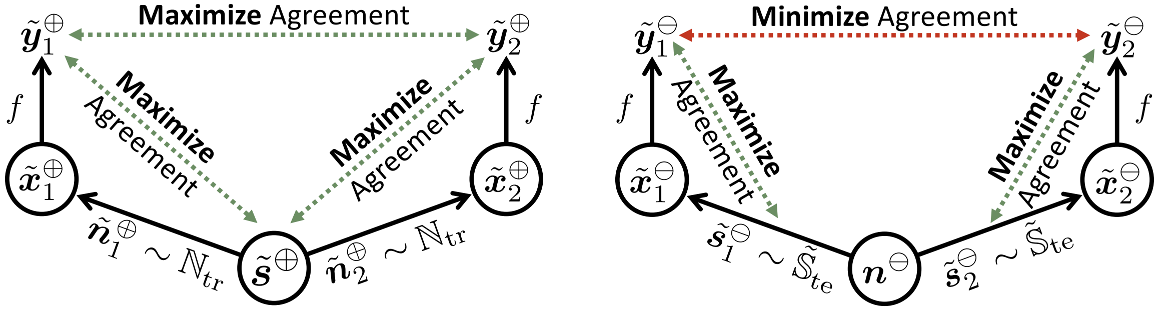
Fig. 3.3 shows a visualization of the PseudoSE pretraining process. The training target is pseudo-clean speech $\tilde{\boldsymbol{s}}$, therefore the model parameters $\mathcal{W}_{\text{PseudoSE}}$ are iteratively updated to minimize the loss function $\mathcal{L}_{\text{PseudoSE}}$. We simulate the 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. After the model parameters $\mathcal{W}_{\text{PseudoSE}}$ are learned, we may apply finetuning using known clean speech from the scarce set $\mathbb{T}_{\mathbf{f}\text{-tr}}$. In this FSL personalization context, the training targets are genuine clean speech utterances $\boldsymbol{s} \in \mathbb{T}_{\mathbf{f}\text{-tr}}$. Therefore, the parameters from the pseudo enhancement function $\mathcal{W}_{\text{PseudoSE}}$ are iteratively updated in order to fit a real speech enhancement function. Once again, the finetuning loss function is equivalent to Eq. (3.2) using the speaker-specific mixtures.

There are trade-offs to note when using the proposed NTT solution. On one hand, the success of PseudoSE pretraining is bounded by the noisiness of $\tilde{\boldsymbol{s}}$, 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.

3.2.2 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 $\tilde{\mathbb{T}}_{\mathbf{p}\text{-tr}}$ and the deformation noise set $\mathbb{N}_{\text{tr}}$ are large, the quality of the features learned through PseudoSE are bounded by how noisy $\tilde{\mathbb{T}}_{\mathbf{p}\text{-tr}}$ 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 $\mathcal{F}$ 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 Fig. 3.4; note that solid lines indicate signal path while dashed lines show loss terms.

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 $\mathcal{F}$ must also satisfy the contrastive objective based on the fact



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

(a) Positive Pairs (b) Negative Pairs

Figure 3.4: The proposed framework for contrastive mixtures.

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:

$$\mathcal{L}_p = \mathcal{E}(\tilde{\boldsymbol{s}}^{\oplus}||\tilde{\boldsymbol{y}}_1^{\oplus}) + \mathcal{E}(\tilde{\boldsymbol{s}}^{\oplus}||\tilde{\boldsymbol{y}}_2^{\oplus}) + \lambda_p[\mathcal{E}(\tilde{\boldsymbol{y}}_1^{\oplus}||\tilde{\boldsymbol{y}}_2^{\oplus})], \tag{3.6}$$

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:

$$\begin{aligned}\mathcal{L}_n = {} & \mathcal{E}(\tilde{\boldsymbol{s}}_1^{\ominus}||\tilde{\boldsymbol{y}}_1^{\ominus}) + \mathcal{E}(\tilde{\boldsymbol{s}}_2^{\ominus}||\tilde{\boldsymbol{y}}_2^{\ominus}) \\ & + \lambda_n\big[\max\big(\mathcal{E}(\tilde{\boldsymbol{s}}_1^{\ominus}||\tilde{\boldsymbol{s}}_2^{\ominus}), \mathcal{E}(\tilde{\boldsymbol{y}}_1^{\ominus}||\tilde{\boldsymbol{y}}_2^{\ominus})\big)\big],\end{aligned} \tag{3.7}$$

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,

$$\mathcal{L}_{\text{CM}} = \sum_{t=1}^{T} \mathcal{L}_p(t) + \sum_{t=1}^{T} \mathcal{L}_n(t) \tag{3.8}$$

$$\mathcal{W}_{\text{CM}} \leftarrow \operatorname*{arg\,min}_{\mathcal{W}_{\text{CM}}} \mathcal{L}_{\text{CM}}, \tag{3.9}$$

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. (3.4). 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 Section 3.3.4.

Our proposed CM approach differs from the SimCLR model [85] 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.

As illustrated in Fig. 3.5, 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 $\tilde{\boldsymbol{s}}_2^{\ominus}$. Model parameters $\mathcal{W}_{\text{CM}}$ are iteratively updated to minimize the loss function $\mathcal{L}_{\text{CM}}$.

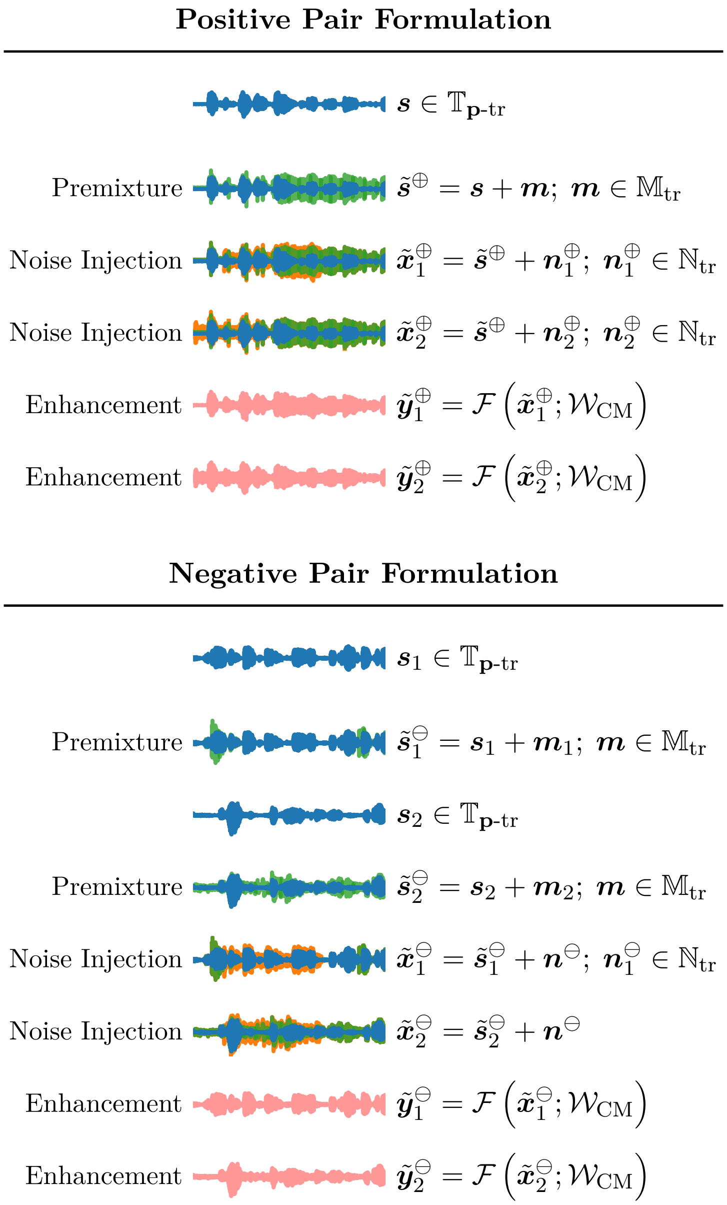

Positive Pair Formulation
$\boldsymbol{s} \in \mathbb{T}_{\mathbf{p}\text{-tr}}$
Premixture
$\tilde{\boldsymbol{s}}^{\oplus} = \boldsymbol{s} + \boldsymbol{m};\ \boldsymbol{m} \in \mathbb{M}_{\text{tr}}$
Noise Injection
$\tilde{\boldsymbol{x}}_1^{\oplus} = \tilde{\boldsymbol{s}}^{\oplus} + \boldsymbol{n}_1^{\oplus};\ \boldsymbol{n}_1^{\oplus} \in \mathbb{N}_{\text{tr}}$
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
$\tilde{\boldsymbol{y}}_1^{\oplus} = \mathcal{F}\left(\tilde{\boldsymbol{x}}_1^{\oplus}; \mathcal{W}_{\text{CM}}\right)$
Enhancement
$\tilde{\boldsymbol{y}}_2^{\oplus} = \mathcal{F}\left(\tilde{\boldsymbol{x}}_2^{\oplus}; \mathcal{W}_{\text{CM}}\right)$
Negative Pair Formulation
$\boldsymbol{s}_1 \in \mathbb{T}_{\mathbf{p}\text{-tr}}$
Premixture
$\tilde{\boldsymbol{s}}_1^{\ominus} = \boldsymbol{s}_1 + \boldsymbol{m}_1;\ \boldsymbol{m} \in \mathbb{M}_{\text{tr}}$
$\boldsymbol{s}_2 \in \mathbb{T}_{\mathbf{p}\text{-tr}}$
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
$\tilde{\boldsymbol{y}}_1^{\ominus} = \mathcal{F}\left(\tilde{\boldsymbol{x}}_1^{\ominus}; \mathcal{W}_{\text{CM}}\right)$
Enhancement
$\tilde{\boldsymbol{y}}_2^{\ominus} = \mathcal{F}\left(\tilde{\boldsymbol{x}}_2^{\ominus}; \mathcal{W}_{\text{CM}}\right)$

Figure 3.5: Single-speaker (self-supervised) contrastive mixtures setup.

3.2.3 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 $\mathbb{S}_{\text{tr}}$. However, the target signals' cleanliness is ambiguous in the case of self-supervised pretraining, which utilizes $\tilde{\mathbb{T}}_{\textbf{p}\text{-tr}}$ as the pseudo source. Based on our formulation of the premixture process in Fig. 3.3, two factors determine whether the pseudo sources $\tilde{\boldsymbol{s}}$ 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 $\tilde{\boldsymbol{s}}$ may contain near-clean speech. Considering all the available noisy utterances $\tilde{\boldsymbol{s}} \in \tilde{\mathbb{T}}_{\textbf{p}\text{-tr}}$, we hypothesize that utterances with a higher SNR may serve as more useful target signals than other noisier utterances, even if none of them 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., $\mathbb{S}_{\text{tr}}$ 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:

$$\text{SegSNR}_j(\boldsymbol{v}, \hat{\boldsymbol{v}}) = 10 \log_{10} \left[\frac{\sum_{i=Hj}^{Hj+N-1} \left(w_{(i-Hj)} v_i \right)^2}{\sum_{i=Hj}^{Hj+N-1} \left(w_{(i-Hj)} r_i \right)^2} \right], \tag{3.10}$$

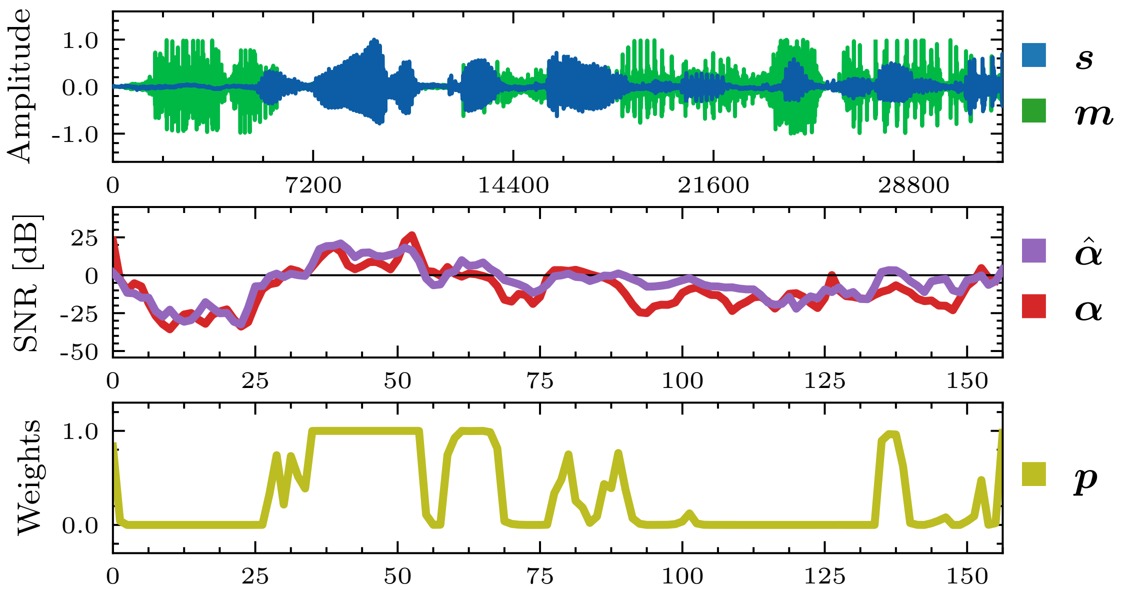


Figure 3.6: Illustration of the SNR predictor inputs and outputs. The first subplot features an example premixture/pseudo source $\tilde{\boldsymbol{s}}$. In the second subplot, the SNR predictor network h estimates the frame-wise (i.e., segmental) SNR of the premixture. The training objective of h is 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. $\boldsymbol{p} = \sigma(h(\tilde{\boldsymbol{s}}))$.

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:

$$\begin{aligned}
&\boldsymbol{x} = \boldsymbol{s} + \boldsymbol{n}; \quad \boldsymbol{s} \in \mathbb{S}_{\text{tr}}, \ \boldsymbol{n} \in \mathbb{N}_{\text{tr}} \\
&\boldsymbol{\alpha} = \text{SegSNR}(\boldsymbol{s}, \boldsymbol{x}) \\
&\hat{\boldsymbol{\alpha}} = h(\boldsymbol{x}; \mathcal{W}_h) \\
\mathcal{W}_h \leftarrow &\operatorname*{arg\,min}_{\mathcal{W}_h} \ \text{MSE}(\hat{\boldsymbol{\alpha}}, \boldsymbol{\alpha}),
\end{aligned} \tag{3.11}$$

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:

$$\boldsymbol{p} = \sigma(\hat{\boldsymbol{\alpha}}) = \frac{1}{1 + e^{-\hat{\boldsymbol{\alpha}}}}. \tag{3.12}$$

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. (3.10) 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 $\overline{\text{SegSNR}}$.

$$\begin{aligned}\mathcal{E}_{\text{DP}}(\tilde{\boldsymbol{y}} \parallel \tilde{\boldsymbol{s}}) &= \overline{\text{SegSNR}}(\tilde{\boldsymbol{y}}, \tilde{\boldsymbol{s}}; \boldsymbol{p}) \\ &= -\frac{1}{J} \sum_{j=0}^{J-1} p_j \left[10 \log_{10} \frac{\sum_{i=Hj}^{Hj+N-1} \left(w_{(i-Hj)} \tilde{s}_i \right)^2}{\sum_{i=Hj}^{Hj+N-1} \left(w_{(i-Hj)} \tilde{r}_i \right)^2} \right]\end{aligned} \tag{3.13}$$

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. 3.6, 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. (3.4)—we obtain a new data purified loss function:

$$\mathcal{L}_{\text{PseudoSE+DP}} = \mathcal{E}_{\text{DP}}(\tilde{\boldsymbol{y}} \parallel \tilde{\boldsymbol{s}}). \tag{3.14}$$

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.

$$\boldsymbol{p}^{\oplus} = \sigma(h(\tilde{\boldsymbol{s}}^{\oplus})),\ \ \boldsymbol{p}_1^{\ominus} = \sigma(h(\tilde{\boldsymbol{s}}_1^{\ominus})),\ \ \boldsymbol{p}_2^{\ominus} = \sigma(h(\tilde{\boldsymbol{s}}_2^{\ominus})) \tag{3.15}$$

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. (3.6) and Eq. (3.7) as:

$$\begin{aligned}\mathcal{L}_{p+\text{DP}} = &\overline{\text{SegSNR}}(\tilde{\boldsymbol{y}}_1^{\oplus}, \tilde{\boldsymbol{s}}^{\oplus}; \boldsymbol{p}^{\oplus}) + \\ &\overline{\text{SegSNR}}(\tilde{\boldsymbol{y}}_2^{\oplus}, \tilde{\boldsymbol{s}}^{\oplus}; \boldsymbol{p}^{\oplus}) + \\ &\lambda_p \big[\overline{\text{SegSNR}}(\tilde{\boldsymbol{y}}_1^{\oplus}, \tilde{\boldsymbol{y}}_2^{\oplus}; \boldsymbol{p}^{\oplus})\big]\end{aligned} \tag{3.16}$$

$$\begin{aligned}\mathcal{L}_{n+\text{DP}} = &\overline{\text{SegSNR}}(\tilde{\boldsymbol{y}}_1^{\ominus}, \tilde{\boldsymbol{s}}_1^{\ominus}; \boldsymbol{p}_1^{\ominus}) + \\ &\overline{\text{SegSNR}}(\tilde{\boldsymbol{y}}_2^{\ominus}, \tilde{\boldsymbol{s}}_2^{\ominus}; \boldsymbol{p}_2^{\ominus}) + \\ &\lambda_n \big[\max\big(\overline{\text{SegSNR}}(\tilde{\boldsymbol{s}}_1^{\ominus}, \tilde{\boldsymbol{s}}_2^{\ominus}; \boldsymbol{p}^{\ominus}), \\ &\qquad\quad \overline{\text{SegSNR}}(\tilde{\boldsymbol{y}}_1^{\ominus}, \tilde{\boldsymbol{y}}_2^{\ominus}; \boldsymbol{p}^{\ominus})\big)\big]\end{aligned} \tag{3.17}$$

The data-purified positive and negative loss functions may now be substituted in Eq. (3.8) to obtain the overall CM+DP loss function:

$$\mathcal{L}_{\text{CM+DP}} = \sum_{t=1}^{T} \mathcal{L}_{p+\text{DP}}(t) + \sum_{t=1}^{T} \mathcal{L}_{n+\text{DP}}(t). \tag{3.18}$$

3.3 Experiment

3.3.1 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. 3.2): 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. (3.2). 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. (3.4). The proposed self-supervised method relies solely on noisy speaker-specific data $\tilde{\mathbb{T}}_{\mathbf{p}\text{-tr}}$.

- **PseudoSE+DP**: Models trained to minimize Eq. (3.14). 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 minimize the feature learning contribution of noise-dominant frames within $\tilde{\mathbb{T}}_{\mathbf{p}\text{-tr}}$.

- **CM**: Models trained to minimize Eq. (3.8). 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. (3.18). 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.

3.3.2 Datasets

Table 3.1 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.

Table 3.1: Glossary of datasets paired with experiment-specific corpora.

Set	**Split**	**Duration**	**Quantity**	**Description**
$\mathbb{S}$	$\mathbb{S}_{\text{tr}}$	443 h	1,132 spkrs	Clean speech from many anonymous speakers
	$\mathbb{S}_{\text{vl}}$	8 h	20 spkrs	
$\mathbb{T}$	$\mathbb{T}_{\mathbf{p}\text{-tr}}$	22.5 min/spkr	20 spkrs	Used to simulate user’s noisy speech which we call “*premixture*” data—$\tilde{\mathbb{T}} = \mathbb{T} \times \mathbb{M}$
	$\mathbb{T}_{\mathbf{p}\text{-vl}}$	60 sec/spkr		
	$\mathbb{T}_{\mathbf{f}\text{-tr}}$	up to 30 sec/spkr		Treated as enrollment data—user-provided scarce clean speech—used only for FSL
	$\mathbb{T}_{\mathbf{f}\text{-vl}}$	30 sec/spkr		
	$\mathbb{T}_{\text{te}}$	30 sec/spkr		Set-aside clean speech from user used only for objective model evaluation
$\mathbb{M}$	$\mathbb{M}_{\text{tr}}$	48 h	13,339 noises	Premixture noises that degrade majority of user’s utterances; unknown to the model
	$\mathbb{M}_{\text{vl}}$	7 h	1,929 noises	
$\mathbb{N}$	$\mathbb{N}_{\text{tr}}$	5 h	616 noises	Injection noises used during model pretraining and fine-tuning
	$\mathbb{N}_{\text{vl}}$	0.5 h	60 noises	
	$\mathbb{N}_{\text{te}}$	0.5 h	60 noises	Injection noises never seen during any model training, used to prepare target speaker-specific test sets

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 [30] for clean speech recordings ($\mathbb{S}$ and $\mathbb{T}$), FSD50K [34] for premixture noises ($\mathbb{M}$), and MUSAN [32] for the injected noises ($\mathbb{N}$).

Out of the LibriSpeech *train-clean-100* subset, we set aside 20 speakers to be the personalization targets; in other words, there are twenty speaker-specific datasets $\mathbb{T}^{(i)}$ where $i \in \{1, \dots, 20\}$. We omit the speaker index i 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 $\mathbb{S}$. 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 $\mathbb{T}$ into five sets as shown in Table 3.1. The utterances are sorted by duration and grouped such that approximately 30 sec are available for testing the model ($\mathbb{T}_{\text{te}}$), 30 sec for validating finetuned models ($\mathbb{T}_{\mathbf{f}\text{-vl}}$), 60 sec for FSL-based finetuning ($\mathbb{T}_{\mathbf{f}\text{-tr}}$), and 60 sec to validate the self-supervised pretraining methods ($\mathbb{T}_{\mathbf{p}\text{-vl}}$). The remaining 22.5 min are used for pretraining ($\mathbb{T}_{\mathbf{p}\text{-tr}}$). Subsequently, for each of the 20 personalization targets, a test set of 100 mixtures is constructed by combining $\mathbb{T}_{\text{te}}$ 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 $\mathbb{S}$ to encompass diverse speaker characteristics. Second, we ensure that the 20 personalization target speakers have enough clean speech signals $\mathbb{T}_{\mathbf{p}\text{-tr}}$ in order to simulate the abundant premixture signals $\tilde{\mathbb{T}}_{\mathbf{p}\text{-tr}}$. 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 [86, 87]. 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 [88]. 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.

With our experiments, we report three metrics frequently used in speech enhancement research: SDR [38], PESQ [42], and extended STOI [41]. 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.

3.3.3 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

Table 3.2: List of model architectures, configurations, and sizes.

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

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 [89, 90, 91]. 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 [49]. 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 $H_c/B_c \approx 4$ [92].

As shown in Table 3.2, 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. 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.

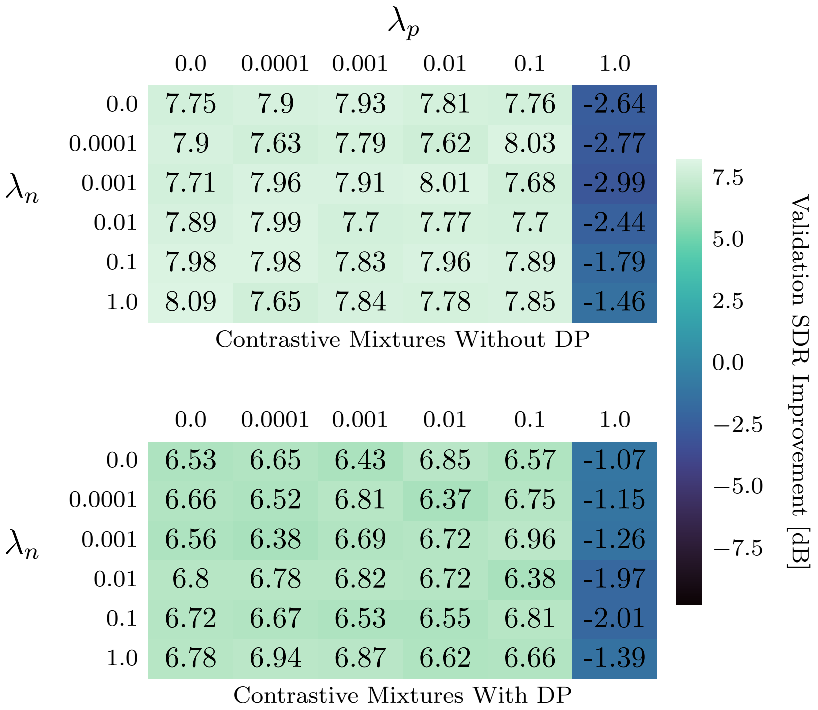
3.3.4 Implementation Details

All models were implemented using PyTorch [93] and trained on NVIDIA Tesla V100 graphics cards. We used the ConvTasNet implementation found in the Asteroid package [94]. All experiments have a fixed batch size of 64. We utilize the Adam optimizer [95] with an initial learning rate of $1\mathrm{e}-3$. When finetuning over clean speech data ($\mathbb{T}_{\mathbf{f}\text{-}\mathrm{tr}}$), the learning rate is instead $1\mathrm{e}-4$. 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.

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. (3.6) and Eq. (3.7) with DP variants Eq. (3.16) and Eq. (3.17). Therefore, we run an ablation study of contrastive mixtures by performing a grid search over potential choices: 1, $1\mathrm{e}-1$, $1\mathrm{e}-2$, $1\mathrm{e}-3$, $1\mathrm{e}-4$, and 0. We can assess the effectiveness of the positive and negative pairs by setting either one of λ_n or 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 3.2. This is done for three out of the twenty personalization target speakers from LibriSpeech. This results in 216 total trials, given that



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

Contrastive Mixtures Without DP

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

Contrastive Mixtures With DP

Figure 3.7: 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.

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. (3.8) or (3.18).

From Fig. 3.7, 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.

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

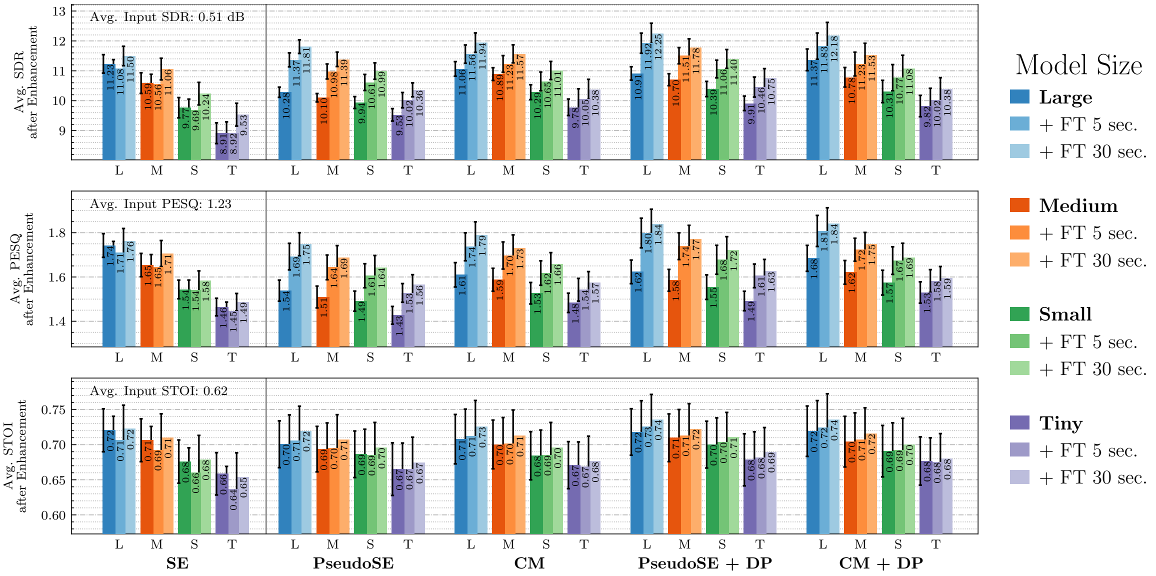


Figure 3.8: noisy-target training (NTT) experiment results.

3.3.5 Results

Next, we discuss the results from the main experiment. As described in in Section 3.3.1, 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.

Fig. 3.8 shows test set results in terms of three signal quality metrics defined in Section 2.2. The improvement for each metric (SDR, PESQ, and STOI) may be calculated by subtracting the average input value from the average value after enhancement. Our results 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. Performances reported using 0 sec represent the ZSL capabilities of the pretraining method. We explore FSL contexts of 5 sec and 30 sec to investigate high and low amounts of data efficiency. The left-most boxplots within the SE column represent unpersonalized / generalist performance. Error bars show the specific 95%-confidence interval per model and training configuration, averaged over all target speakers.

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.

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.

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 ($\tilde{\mathbb{T}}_{\mathbf{p}\text{-tr}}$ 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.

Impact of DP with PseudoSE As shown in our prior work [23], DP can identify cleaner frames from premixture signals $\tilde{\mathbb{T}}_{\mathbf{p\text{-}tr}}$ 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 dB (10.91 vs. 10.28) in the large models, while the tiny models only see an average boost of about 0.38 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 1.0dB improvement from the baseline SE model (9.91 vs. 8.91).

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.

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.

Model Compression Among the tiny-sized models, the best-performing ZSL method for personalization is **PseudoSE+DP** which produced an average SDR improvement of 9.91dB. We see that the personalized tiny model outperforms the generalist small model (9.77dB), although it uses 62% fewer model parameters and multiply-accumulate operations (MACs) according to Table 3.2. 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 (10.89dB) has its confidence interval overlapped with that of the largest SE baseline (11.23dB), 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.

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 dB for the tiny-sized models or 0.7 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.

FSL Personalization Performance

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

Generalist Models' Performance We observe that all four sizes of the baseline models pretrained as generalists (**SE**) are incapable of adapting over a small $\mathbb{T}_{\texttt{f-tr}}$ that has only 5 sec of data. Using 30 sec of clean speech data does eventually produce gains for all model sizes. The tiny-sized generalist sees the most significant gains (0.62 dB) whereas the large-sized generalist sees marginal benefit (0.27 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.

FSL after PseudoSE Initialization We reiterate that our self-supervised methods train using noisy speaker-specific data with premixture SNRs in the 0 dB to 15 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 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 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 sec of clean speech for finetuning achieve a greater mean SDR improvement compared to generalists models finetuned using 30 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.

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.

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 sec of clean speech to get 11.06 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 dB). This example shows a lossless 78% reduction in model parameters and MACs.

3.3.6 Summary

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.

Chapter 4

Personalization through Test-Time Model Selection

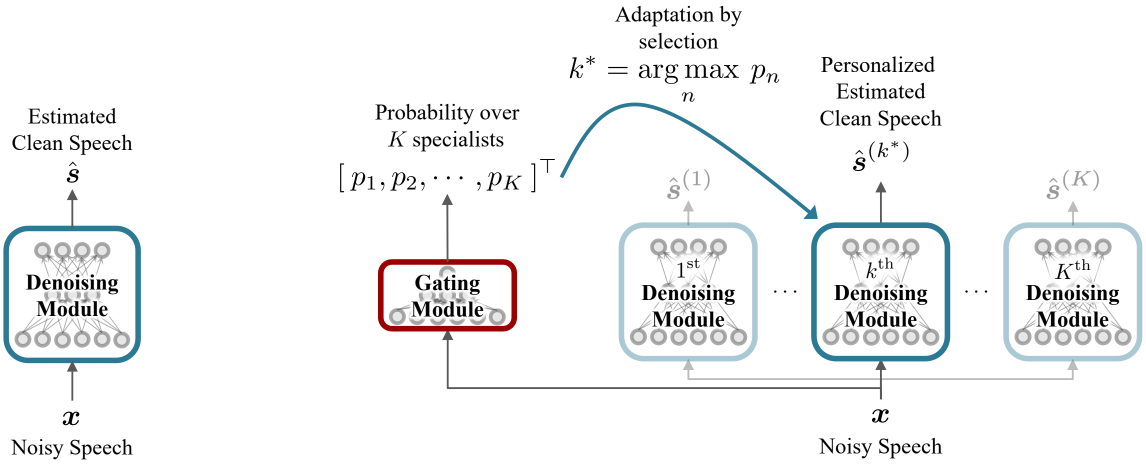
In this chapter, we investigate how the concept of model selection can be utilized for zero-shot adaptation of speech enhancement systems.

Specifically, we leverage the mixture of local experts (MLE) paradigm in the deep learning context, particularly because the design enables a reduction in computational cost. While most ensemble machine learning techniques combine the outputs of its “weak learners”, we are instead interested in using a few—or at best, one—learner to achieve speedier inference. The main insight is that the overarching problem space may be divided into homogeneous regions (thus the name “local expert”). While prior works have shown that the speech enhancement problem can also be divided in some manner, our contribution emphasizes on adaptation (the improved performance with particular speakers or in certain environments) in relation to the reduced computational complexity. Finally, we show that model selection realizes zero-shot adaptation since the training data may be constructed without any knowledge of the test-time speaker or environment.

4.1 Sparse Ensemble of Specialists

Given that the speech denoising task can be divided into mutually exclusive subproblems, we propose that it must be possible to split a complete noisy speech dataset along some latent dimension in order to form non-overlapping subsets (i.e. clusters). Although the MLE procedure is theoretically capable of learning latent clusters in an unsupervised fashion, for our initial experiments, we incorporate our prior knowledge about the problem domain to manually define latent spaces that subdivide the speech enhancement problem. These include: (1) different speech degradation levels and (2) speaker gender.

The proposed model, shown in Fig. 4.1b, is an ensemble of specialist networks regulated by a gating network. While it is fundamentally possible to utilize the inferences of multiple specialists, we propose using only a single specialist in order to bring computational complexity during inference to a minimum. We assume that the noisy speech data can be split into distinct subsets. Consequently, we pre-train each specialist network to individually address one subproblem. Our experiments compare the proposed ensemble model against a baseline model, shown in Fig. 4.1a, which is architecturally equivalent to a single specialist network but is trained using the entire noisy speech training set. Next, we define the specialist and gating modules more formally.



(a) (b)

Figure 4.1: Comparison between (a) the typical non-ensemble denoising model, and (b) the proposed sparse ensemble of specialists.

4.1.1 Specialist Networks

With consideration for the constraints of resource-limited environments, we design our specialist network with unidirectional recurrent layers followed by a feed-forward dense layer. The recurrent layers are made up of long short-term memory (LSTM) cells [96]. The number of recurrent layers as well as the number of hidden units per layer are adjustable experiment parameters which affect the overall complexity of the model. The specialist network takes the noisy speech magnitude STFT $|\boldsymbol{X}|$ as input and predicts a ratio mask matrix $\boldsymbol{M}$. Subsequently, inv-STFT $(\boldsymbol{M} \odot \boldsymbol{X})$ yields the denoised speech estimate $\hat{\boldsymbol{s}}$.

We note that convolutional neural networks (CNNs) on time-domain signals currently achieves improved performance in source separation [49]. Despite their low model complexity, convolutional architectures are able learn the sequence-to-sequence mapping. We leave general application of our proposed ensemble model to different architectures for future work.

4.1.2 Gating Network

The gating network is responsible for assigning an input signal to the appropriate specialist. It introduces a classification sub-task as overhead to the overarching denoising task, splitting the full training dataset into some number of latent clusters.

Identifying latent clusters in a noisy speech corpus is non-trivial. Prior works using ensemble models for speech enhancement have shown that specialists may be trained to denoise a particular phoneme [97]. This approach, which requires training data to be phoneme-labeled, is naturally language-dependent but also non-sparse, as multiple specialists may actively perform some computations due to the high variance of phonemes in speech. To ensure a sparse activation of specialists (ideally one specialist per input signal), a more generalized latent clustering is preferred. For this reason, we design two types of gating networks to classify inputs based on either *speech degradation level* or *speaker gender*.

Similar to the specialist architecture, our gating networks are also designed with multiple recurrent layers and a single dense layer. However, in our current proposed model the gating network does not make predictions frame-by-frame; after processing the entire input sequence, the network produces a single softmax vector $\boldsymbol{p}$, with K elements corresponding to the number of clusters (i.e. the number of specialists). The index of the maximum value in $\boldsymbol{p}$ should correspond to the index of the best-suited specialist.

4.1.3 Ensemble Network

The proposed ensemble model combines K specialist networks together with a gating network. First, all of the sub-networks are independently trained. The combination of these pre-trained modules forms a primitive ensemble, as the gating network can already assign an incoming test example to one of the specialists. The output mask $\boldsymbol{Y}$ is chosen from the specialist which corresponds to the maximum value of gating network softmax vector $\boldsymbol{p}$. The "hard" gating mechanism is formulated as:

$$\boldsymbol{Y} = \boldsymbol{M}^{(k^*)}, \quad k^* = \arg\max_k p_k, \tag{4.1}$$

where $\boldsymbol{M}^{(k)}$ denotes the predicted ratio mask matrix from the k-th specialist.

However, this naïve ensemble is sub-optimal as it lacks the potential co-adaptation between gating and specialist networks. For example, given the fact that the gating network cannot classify mixtures with 100% accuracy, the specialists should adapt to the situation where it processes a misclassified sample (e.g., a male speech sample falls in the female speaker's specialist). Knowing this, we can further train the submodules in unison. During this fine-tuning phase, the ensemble model estimates the output ratio mask $\boldsymbol{M}$ by performing a normalized sum over the individual masks $\boldsymbol{M}^{(k)}$ produced by all specialists weighted by the gating network softmax vector $\boldsymbol{p}$. This "soft" gating mechanism ensures that the ratio mask calculation is differentiable, and is formulated as:

$$\boldsymbol{Y} = \sum_{k} p_k \boldsymbol{M}^{(k)}. \tag{4.2}$$

During the test phase, the weighted sum is replaced by the hard-decision shown in Eq. (4.1). This difference between training-time and evaluation-time computation in the ensemble architecture is the crux of its efficiency; only one out of all the specialists is used to process the entire mixture spectrogram $|\boldsymbol{X}|$, making the total used network parameters a fraction of the total learned. We reduce the discrepancy between the hard and soft gating mechanisms, used during testing and fine-tuning respectively, by introducing a scaling parameter λ to the softmax gating network output:

$$p_k = \frac{\exp(\lambda \cdot o_k)}{\sum_{j=1}^{K} \exp(\lambda \cdot o_j)}. \tag{4.3}$$

Each element of the gating network output cluster probability vector (p_k) is dependent on the corresponding element of dense layer output (o_k) normalized by the sum of all dense layer output elements. While the traditional softmax function can be calculated using $\lambda = 1$, we elevate the sparsity of $\boldsymbol{p}$ by setting $\lambda = 10$. This saturates $\boldsymbol{p}$ to be near-1 at a single index and near-0 at every other index, making the weighted sum for ratio mask $\boldsymbol{M}$ (Eq. (4.2)) effectively select the best-case specialist mask. This modification of the softmax function has been successfully used for quantizing vectors with image compression [98].

4.1.4 Experiment Setup

All models (specialist, gating, baseline, and ensemble) are trained using a stochastic data sampling strategy which dynamically mixes clean speech recordings from the LibriSpeech[1] corpus [30] with noise recordings from the MUSAN[2] corpus [32]. This exposes the models to

---

[1] Available for download at `http://www.openslr.org/12/`.

[2] Available for download at `http://www.openslr.org/17/`.

up to 251 unique speakers[3] and 843 unique noise types[4] during training. 40 unseen speakers[5] and 87 unseen noise types[6] are used to test the models. 5% of the training utterances and noises are set aside for validation to help determine training convergence.

All experiment audio files use a sampling rate of 16 000 Hz. Spectrograms are generated using the STFT with a frame size of 1024 samples and a hop size of 256 samples. Per epoch, for each example in the training batch, the sampler mixes a normalized 1-second snippet of a random training speaker's utterance with a normalized 1-second snippet from a random training noise, chosen with uniform probability. There are 100 mixture signals in a batch. Unlike the training mixtures, test mixtures vary in duration; this gives our models an effective RNN lookback size of 1-second.

We assess the proposed ensemble of specialists methodology across two latent spaces. For the *signal degradation* latent space, we instantiate $K = 4$ specialists and generate noisy speech mixtures with specific signal-to-noise ratio (SNR) levels—either -5 , 0 , 5 , or 10 dB—for each of the four specialists. Similarly for the *speaker gender* experiment, there are $K = 2$ specialists which see a gender-filtered subset of the training data with uniformly varying input SNR values out of the four above listed. In contrast, the baseline model must generalize to all levels of signal degradation and all speaker genders; its training batches consist of 100 mixed gender 1-second-long mixtures with input SNR uniformly distributed between the four values.

All networks are optimized using the Adam optimizer [95] with an initial learning rate of $\eta = 0.001$. The specialist network uses the additive inverse of the SISDR metric, i.e. Eq. (2.3), between $\hat{\boldsymbol{s}}$ and $\boldsymbol{s}$ as the loss function, whereas the gating network minimizes the binary cross entropy (CE) metric between its output, softmax vector $\boldsymbol{p}$, and a ground-truth

---

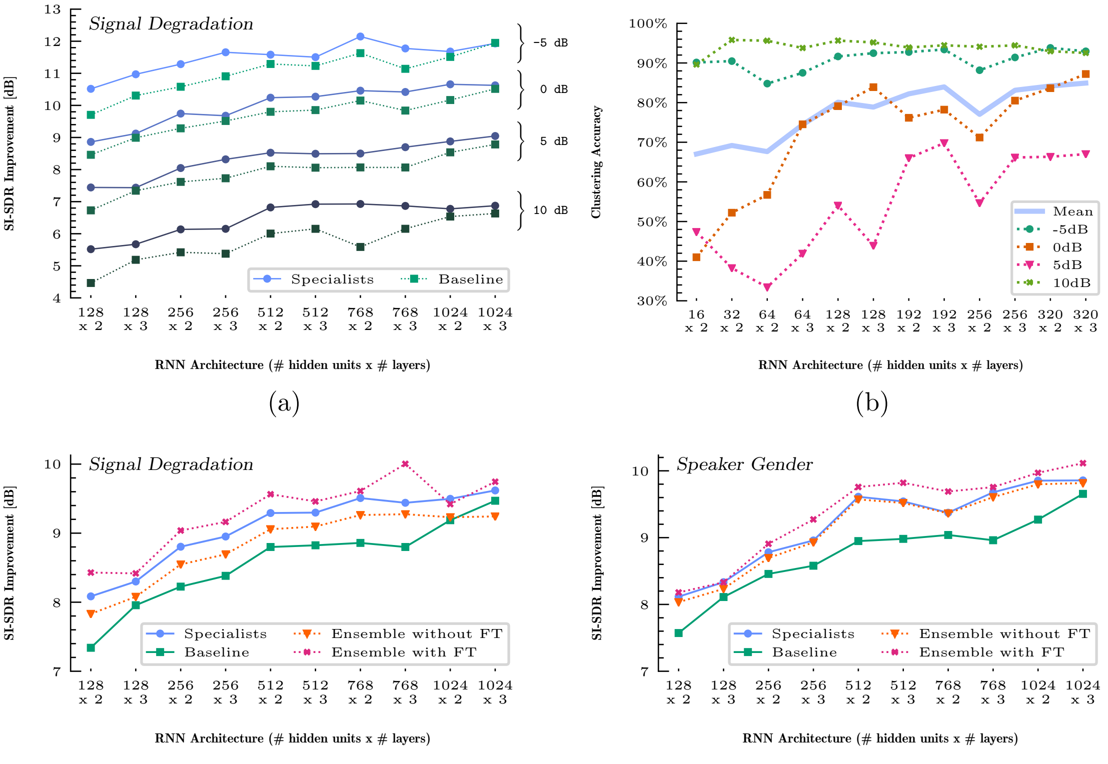
[3] From the `librispeech/train-clean-100` folder.

[4] From the `musan/noise/free-sound` folder.

[5] From the `librispeech/test-clean` folder.

[6] From the `musan/noise/sound-bible` folder.

one-hot vector representing the index of the best-suited specialist, i.e, Eq. (2.5). Each network variant is trained for approximately three hours on a NVIDIA Titan Xp GPU, after which the validation metric is considered to have converged.



(a)

(b)

(c)

(d)

Figure 4.2: Results from the signal degradation and speaker gender experiments. The LSTM component of the specialist network increases in computational complexity going across the x-axis on all subplots.

4.1.5 Results

We report the denoised signal SISDR improvement for all models averaged across 1000 test set mixtures. Fig. 4.2a compares the test signal speech denoising performance between the four *signal degradation*-based specialists and the one baseline model. It is evident that, at all mixture SNR levels, a neural network specifically trained to denoise those mixtures can outperform a generalist network. This gap in performance is most prominent with the extrema mixture levels (i.e., the $-5\,\text{dB}$ and $10\,\text{dB}$ mixture SNR cases). As the number of

RNN hidden units and layers increases, the performance gap between specialists and baseline model diminishes. With larger network complexity, the generalist's performance eventually matches the specialist's, which saturates after a particular network size.

The specialist curves in Fig. 4.2a, 4.2c, Fig. 4.2d set a theoretical upper bound to the naïve ensemble model: even with a perfect gating network, the naïve ensemble cannot outperform the sum of its parts. The superior performance of the naïve ensemble model to the baseline comes from the fact that each specialist focuses on a smaller subset of the original problem with the same model capacity. In this hypothetical context where the best-suited specialist is always selected, an ensemble of smaller specialist networks will consistently outperform the baseline generalists.

Therefore, the gating network's classification accuracy matters. As shown in Fig. 4.2b, *signal degradation*-based gating networks with a smaller RNN architecture are only able to distinguish the extrema mixture levels with high confidence. Increasing the number of hidden units and layers brings up the classification accuracy of the non-extrema mixture levels (i.e. 0 dB and 5 dB mixture SNR). Based on these results, we chose the 128×2 gating network architecture to be used for the subsequent ensemble experiments, as it adequately clusters test mixtures (with $\approx 80\%$ accuracy on average) while only incurring a small computational overhead.

Fig. 4.2c compares the averaged denoising performance of the individual specialists, the baseline, and the ensemble models (with and without fine-tuning) across all four mixture SNR cases. We can see that the naïve ensemble improves upon the baseline with a significant margin, but cannot pass the theoretical upper bound set by the oracle choice of specialist. Still, the naïve ensemble model can compete as an efficient inference model with the high-complexity baseline model of size 1024×2 with a simpler architectural choice, 512×2.

Fig. 4.2c also shows that the fine-tuning step greatly improves our ensemble model, surpassing the oracle specialist upper bound. This suggests that through fine-tuning, the specialists learn to compensate for imperfect classification results from the gating module. We can see that a fine-tuned ensemble with a smaller specialist RNN architecture, 512×2, outperforms the most complex baseline model of size 1024×3. This is a significant amount of computation reduced during the test time, even considering the overhead cost of the 128×2 gating network.

A similar trend is present in the *speaker gender* experiment, summarized in Fig. 4.2d. Since this setup consists of only two specialists, the gating network's job is an easier binary classification. A 16×2 RNN architecture sufficiently classifies speaker gender with 90% classification accuracy. Using that, the naïve ensemble achieves near-optimal performance, reaching the upper bound in nearly every architecture. The fine-tuning process lifts the performance even further.

4.1.6 Summary

With our experiment in this Section 4.1, we demonstrated that speech denoising neural networks can benefit from the MLE design philosophy, boosting performance while reducing arithmetic complexity. Our specialist networks were trained on specific partitions of a large noisy speech corpus across two latent spaces: *signal degradation* and *speaker gender*. Despite the small overhead cost of a gating network, a naïve ensemble network is shown to match the performance of generalist denoising networks with fewer parameters i.e. fewer inference-time calculations. Furthermore, fine-tuning the ensemble with the inclusion of a sparsity parameter helps the model exceed the theoretical upper bound of the oracle specialist.

4.2 Speaker-Informed Sparse Ensemble of Specialists

Now, we will investigate using MLE as a means for *personalizing* an SE model. To achieve this, we propose learning the optimal speaker grouping from the noisy utterances. This is in contrast to the ensemble model of Section 4.1, which operated on manually-defined semantic speaker groups (e.g., input SNR or speaker gender). Using learned speaker groups, the gating module must estimate characteristics of the test-time speaker from the noisy input, identify the most similar speaker group defined within the training set, then forward the input signal to the appropriate specialist network. This schema requires no training data from the test-time speakers, yet it more optimally denoises the test-time noisy utterances by using the most suitable specialist. With this in mind, our proposed model encapsulates "zero-shot" PSE through model selection.

A major aspect of this work addresses the open-ended question: how do we cluster English speakers into appropriate groups? A relevant task is learning speaker-characteristic embeddings for speaker verification (SV) systems. Well-established embeddings include the Gaussian mixture model-based *i-vectors* [99] or *x-vectors* computed using a time-delay neural network [100]. Prior works have also used sequence summarizing networks [101] either through contrastive loss [102] or by estimating subsequent frames for a single input signal [103]. Although these learn valid speaker-identifying features, we propose a custom embedding-learning model which can effectively function as the gating network as in [78]. Additionally, we want our custom embedding to be robust to additive noise; previously proposed noise-robust embedding vectors [104, 105, 65] were not designed around MoLE. To do this, we develop a Siamese network [106], intended for speaker verification (SV), to learn discriminative speaker embeddings. We then repurpose the SV module as a classifier. Through fine-tuning, the ensemble model morphs the learned embedding space from SV-applicable into something more suitable for the SE task. Lastly, because this work utilizes soft gating at training-time and hard gating at test-time [21], our zero-shot sparse ensemble model for personalized SE minimizes test-time computational complexity.

4.2.1 Design

Given a large dataset of many different speakers' various utterances $\mathbb{S}$, we postulate that there exists an optimal clustering based around speaker identifying characteristics. Denoting K to be the number of clusters, one can create K separate SE models trained only to denoise utterances from each disjoint group of similar speakers. As previously shown [21, 79], a sparsely active ensemble model is capable of performing zero-shot adaptation because the gating module classifies the test-time *noisy* utterances into one-of-K groups.

An ensemble model is composed of one gating module and K specialist modules. The gating module processes a noisy speech input frame $\boldsymbol{x}$, estimating a speaker-embedding first, and then classifying it as belonging to one-of-K groups. The cluster probabilities vector $\boldsymbol{p}$ is used in two ways—during training, all of the specialist modules outputs their own ideal ratio mask (IRM) [107] estimates, $\boldsymbol{M}^{(1)}, \boldsymbol{M}^{(2)}, \ldots, \boldsymbol{M}^{(K)}$, which are then combined in a weighted sum using $\boldsymbol{p}$, i.e., $\hat{\boldsymbol{M}} = \sum_{k=1}^{K} p_k \boldsymbol{M}^{(k)}$. But during testing, only the output from the k^*-th specialist, corresponding to the largest probability, i.e., $k^* = \arg\max_k p_k$, is chosen. This argmax operation selects a single specialist to use during evaluation, making the ensemble sparsely active.

In the context of personalized speech enhancement, increasing hyper-parameter K can theoretically increase the level of specialization of each specialist as well as the ensemble network's capacity for personalization. However, there is a trade-off with having too many models; a large K can make the gating module's classification task too challenging, and may lead to the specialist modules becoming overfit on subsets that are too small. In this paper, we investigate three choices of K: 2, 5, and 10. Determining the optimal number of clusters is an extended research topic within unsupervised learning.

4.2.2 Pretraining Process

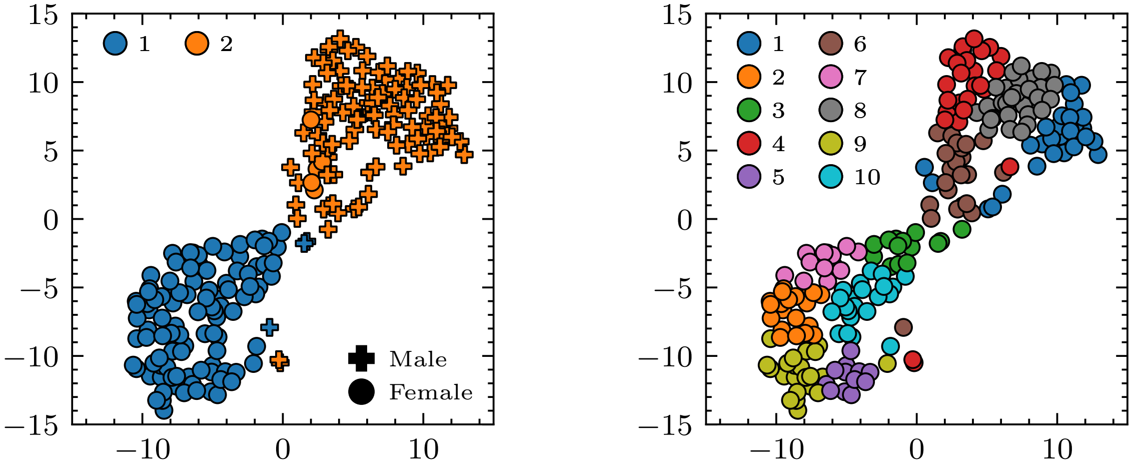
Discriminative Speaker-Specific Embeddings The clustering of speakers is a significant matter when we build a successful sparse ensemble model for SE. Although in theory all the specialists and the gating module can be trained from scratch, training many modules simultaneously is prone to result in suboptimal performance. Hence, we first pre-train all the modules individually and then fine-tune them. The pre-training step, therefore, requires the sub-grouping of speakers.

To this end, we train a neural encoder that learns an embedding function f which can characterize a noisy speech utterance with a low-rank embedding vector. In order to train f, we formulate a speaker verification (SV) upstream task. First, we sample utterances from a large training dataset containing many speakers, $\boldsymbol{s} \in \mathbb{S}$, and noise signals from a similarly large dataset of diverse noises, $\boldsymbol{n} \in \mathbb{N}$. Input mixtures $\boldsymbol{x}$ are made by artificially mixing clean speech utterances $\boldsymbol{s}$ with training noise signals $\boldsymbol{n}$; the amplitude of $\boldsymbol{n}$ is scaled to simulate various signal-to-noise ratios (SNRs).

We can then generate pairs of noisy speech utterances, $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. Once f predicts the embeddings, i.e., $\boldsymbol{z}_i = f(\boldsymbol{x}_i)$ and $\boldsymbol{z}_j = f(\boldsymbol{x}_j)$, their inner product serves as a measure of similarity. A sigmoid function follows to interpret it as a probability $\hat{y}$. Our target is a binary value y, either 1 or 0 depending on whether the utterances derive from the same speaker or not. The embedding function f is trained to minimize the binary cross entropy loss between $\hat{y}$ and y.

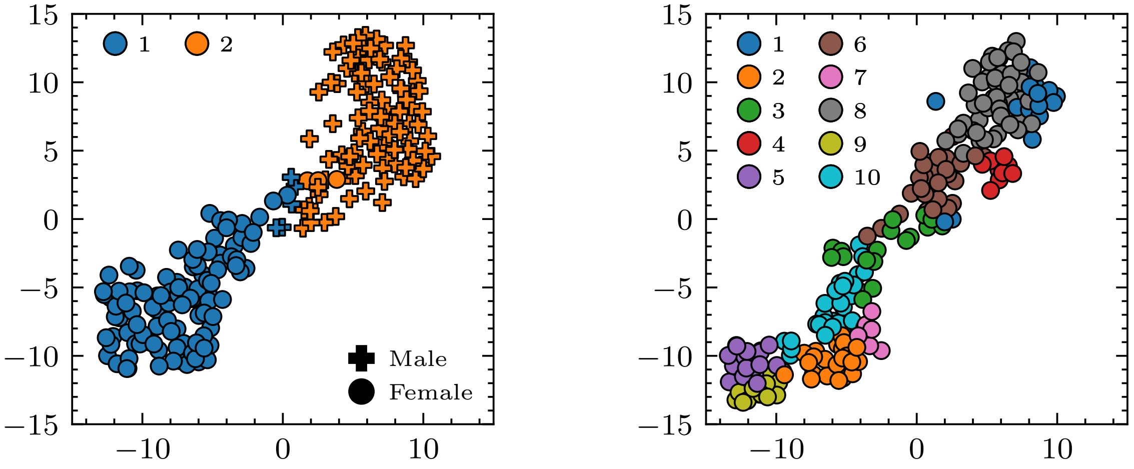
This contrastive learning approach derives discriminative embeddings using Siamese networks [106] where the same embedding function f is applied to both input signals $\boldsymbol{x}_i$

and $\boldsymbol{x}_j$. The rationale behind this embedding model is that the discriminative nature of these embeddings can help the clustering process prepare a semantically more meaningful partitioning of speakers.



(a) Speaker means obtained from SV with $K = 2$ clustering.

(b) Speaker means obtained from SV with $K = 10$ clustering.



(c) Speaker means derived uniquely by a fine-tuned $K = 2$ ensemble.

(d) Speaker means derived uniquely by a fine-tuned $K = 10$ ensemble.

Figure 4.3: Subplots comparing various choices of K for using k-means clustering on the speaker embeddings.

Offline Speaker Clustering Likewise, the gating module's classification task and pre-training of individual specialists rely on a reasonable clustering of speakers. Determining how the K groups are formulated, and which of the training set speakers belongs to each group, requires an offline clustering step. First, we transform every utterance from the training corpus into the learned latent space, i.e., $\boldsymbol{z} \leftarrow f(s)$. Embedding vectors from the same speaker are averaged element-wise, which serves as the speaker-characteristic mean vector. Finally, we apply k-means clustering to these mean vectors to learn K speaker groups.

Fig. 4.3 shows the clustering results with varying K. Each of the 211 points represents one of the Librispeech training set speakers, with marker style indicating speaker gender. For plotting, the 32-dimensional embeddings $\boldsymbol{z}$ are reduced to 2 dimensions using t-SNE (with perplexity = 40) [108]. These subplots show that the SV model succeeds in learning a speaker embedding which can be clustered into loosely meaningful groups, e.g., when $K = 2$ the clusters implicitly form along the speaker gender division. These speaker groups are used to pre-train our gating modules and local experts.

The speaker verification (SV) pre-training task creates a latent space of speaker embeddings $\mathbb{Z}$, from which we can partition various groups, i.e. 2 in Fig. 4.3a and 10 in Fig. 4.3b. After fine-tuning an ensemble model, the gating network's embedding function f adjusts its parameters towards the SE objective. The latent space is modified uniquely based on the ensemble's configuration. In Fig. 4.3a and Fig. 4.3b, the class labels derive from k-means clustering, but in Fig. 4.3c and Fig. 4.3d the class labels are estimated by the gating network's classifier function g.

Gating Module Pre-Training The gating module must be able to classify the embedding vectors as belonging to one of the K speaker clusters. This neural network is a dense layer followed by the softmax activation, which we denote by a parametric function $\boldsymbol{p} = g(\boldsymbol{z}; \mathcal{W}_g)$, where $\mathcal{W}_g$ is its parameters. The classifier function g takes embeddings of noisy utterances $\boldsymbol{z}$ as inputs, and outputs a vector of cluster probabilities $\hat{\boldsymbol{p}}$. As each utterance belongs to a single cluster and the speaker IDs of the training set speakers are known, we can encode the k-means clustering labels into one-hot vector targets $\boldsymbol{p}$. These vectors are K-dimensional.

Note the discrepancy between the clustering done on embeddings of the clean speech utterances and the actual use-case of the model that takes noisy utterances. While the clustering results on clean data might be more reliable, eventually it is always possible that a noisy test utterance can be misclassified into a wrong speaker group, and then consequently assigned to a sub-optimal specialist. Moreover, since the embeddings are optimized for the SV tasks, clustering on this representation may not be optimal for our SE problem. We revisit this issue in Section 4.2.2 and propose a fine-tuning solution.

Specialist Pre-Training The K specialist modules are trained to denoise speech as follows: the large dataset of training noises $\mathbb{N}$ is retained, but the large speech corpus $\mathbb{S}$ is partitioned into K groups, $\{\mathbb{S}^{(1)}, \ldots, \mathbb{S}^{(K)}\}$, based on the clustering results in Section 4.2.2. The k-th specialist module learns a mapping function h by updating its parameters $\mathcal{W}_h$ such that the distance $\mathcal{E}$ between the denoised estimate signal $\hat{\boldsymbol{s}}$ and the target clean speech signal $\boldsymbol{s}$ is minimized. We use the negative scale-invariant signal-to-distortion ratio (SISDR) [39] as the loss function.

Ensemble Fine-Tuning The ensemble model can now be used naïvely by assembling the pre-trained specialist modules and a pre-trained gating module. However, the gating module may not classify all input signals with perfect accuracy. Therefore, fine-tuning (FT) can adjust the ensemble model's denoising performance for misclassified inputs. This potential co-adaptation between gating and specialist modules can be found by adjusting the parameters of all the underlying functions (i.e., embedding function f, classifier function g, and denoising functions h within each specialist). In the fine-tuning phase, the ensemble model estimates the final ratio mask $\hat{\boldsymbol{M}}$ by performing a normalized sum over the individual masks $\boldsymbol{M}^{(k)}$ using the softmax vector, $\hat{\boldsymbol{p}}$, i.e., $\hat{\boldsymbol{M}} = \sum_{k=1}^{K} \hat{p}_k \boldsymbol{M}^{(k)}$. This ensures that the ratio mask calculation is differentiable and can be seen as a "soft" gating mechanism.

During testing, the weighted sum is replaced by a hard-decision, i.e. $\hat{\boldsymbol{M}} = \boldsymbol{M}^{(k^*)}$ where $k^* = \mathrm{argmax}_k\, p_k$. This switch in gating mechanism between training- and evaluation-time is the essence of the ensemble scheme's efficiency: only one out of all the specialists is active during inference, making the total used network parameters a fraction of the total learned. In order to reduce the discrepancy between the hard and soft gating mechanisms (i.e, to make the gating network more sparse during training), we modify the base of the softmax function to use e^{10} as opposed to simply e [21].

Fig. 4.3c and Fig. 4.3d show the fine-tuned speaker embedding vectors. Note that the comparison between the clustering on the SV embedding vectors and on their fine-tuned version is not to argue that fine-tuning can improve the clustering results. Instead, fine-tuning with the speech enhancement objective could in fact deteriorate the discriminative qualities of the learned embedding vectors.

4.2.3 Experiment Setup

Mixtures are generated by combining randomly offset 5 sec segments of utterances and noises. With every mixture, the noise signal is randomly scaled such that the mixture SNR lies uniformly between -5 to 10 dB. Utterances derive from the LibriSpeech corpus [30] *train-clean-100* folder, with 211 speakers designated in the training set, 20 in the validation set, and 20 in the test set. Noises are selected from the MUSAN corpus [32], with 628 noises from the *free-sound* folder used during training and validation, and 54 noises from the *sound-bible* folder used during test. Both LibriSpeech and MUSAN corpora are resampled to 8 kHz. When training the speaker verification model, batches are made up of pairs of mixtures, with an equal chance of being from the same speaker or not. All mixture signals are processed in the time-frequency domain through STFT using a frame size of 1024 samples with 75 % overlap. Throughout our experiment, every model performs speech denoising by taking a series of magnitude spectra as input and estimating IRM vectors $\boldsymbol{M}$. Masking is done element-wise onto the complex-valued spectrum which possesses the noisy phase of the mixture signal.

Both the gating and specialist modules are composed of gated recurrent units (GRU) cells [109]. The embedding function f is built with 2 hidden layers and 32 hidden units, with the output from last frame becoming a fixed-length utterance-characteristic embedding $\boldsymbol{z}$. The denoising functions h are also built with 2 hidden layers but with a varied number of hidden units. The baseline general-purpose SE model is constructed in exactly the same manner as a specialist network, but is trained on the entire speech corpus $\mathbb{S}$ instead of a personalized subset $\mathbb{S}^{(k)}$. Throughout the experiment, we opt for a batch size of 128, training all models using the Adam optimizer with learning rates of 10^{-3} for training and 10^{-4} for fine-tuning.

4.2.4 Results

Fig. 4.4 summarizes the findings of our experiments. The x-axis shows the varying hidden sizes for the GRU layers. Since the number in parenthesis reports each expert's size, the total size of the ensemble model is computed by multiplying K to it, e.g., when $K = 5$ and the hidden size is 256, the total number of parameters equals 5.6 M. However, because our ensemble models are sparsely active—that is, one specialist is active at a time—the number of parameters effective at run-time is only $1/K$ of the total, the amount listed on the x-axis. Longitudinally, the baseline models share the same number of hidden units with the specialist module, meaning the baseline is always K times smaller than the ensemble model in comparison. However their effective number of parameters is nearly equivalent. We note that ensemble models are not fine-tuned for hidden sizes ≥ 512 due to GPU memory constraints. Larger baseline models are trained and evaluated for comparison with the smaller ensemble models.

Firstly, we see that across all configurations, our ensemble models consistently yields a higher denoising performance when compared to a baseline generalist model whose size is similar to one of the specialists. The naïve ensemble models already show significant improvement (ranging from 0.62 to 1.65 dB), but different choices of K do not make a big difference. We also observe that fine-tuning the ensemble models lift the performance even further (from 1.24 to as much as 2.04 dB. Furthermore, fine-tuning introduces a larger gap in improvement when K is larger; intuitively, the more challenging classification task stands to benefit most from fine-tuning.

The proposed method also performs model compression without sacrificing the denoising performance. Overall, the smaller model architecture receives more performance improvement, such as the 2.0 dB improvement in the case of 64 hidden units. The model compression benefits are made clear by comparing data points laterally. For example, as circled in Fig. 4.4, a generalist model requires at least 512 hidden units in order to match the performance of a fine-tuned ensemble model with 10 specialists each made up of GRUs with only 64 hidden units. Including the cost of the gating module and all the other specialists that are not chosen, this is still a 48% reduction in terms of spatial complexity. Moreover, if we only count the gating module and one chosen specialist, it is a 94% reduction in effective parameters and test-time arithmetic complexity.

Lastly, as hypothesized, we see that increasing the number of clusters results can result in a more personalized speech enhancement so long as the ensemble model is fine-tuned. The average SISDR improvement achieved with the ensemble models increases along with K from 2 to 5 to 10 through fine-tuning.

4.2.5 Summary

With this section, we expanded upon model adaptation through selection (the “mixture of local experts” paradigm) as a means for personalized speech enhancement. We show that the speaker-informed ensemble is a zero-shot PSE system as it never requires clean speech during the test-time adaptation; instead, the gating module analyzes the noisy test signal to determine the most appropriate specialist, or local expert, for denoising. We obtain a speaker-informed gating module by pre-training it with a contrastive speaker verification task. The training cases are transformed to a learned latent space where they are clustered using k-means clustering. By identifying more clusters and training more low-cost specialists, our ensemble models are able to adapt better to unseen test environments. Our findings reinforce the idea that sparse ensemble models can outperform general-purpose speech denoising models of a similar architecture, additionally reducing run-time computational complexity.

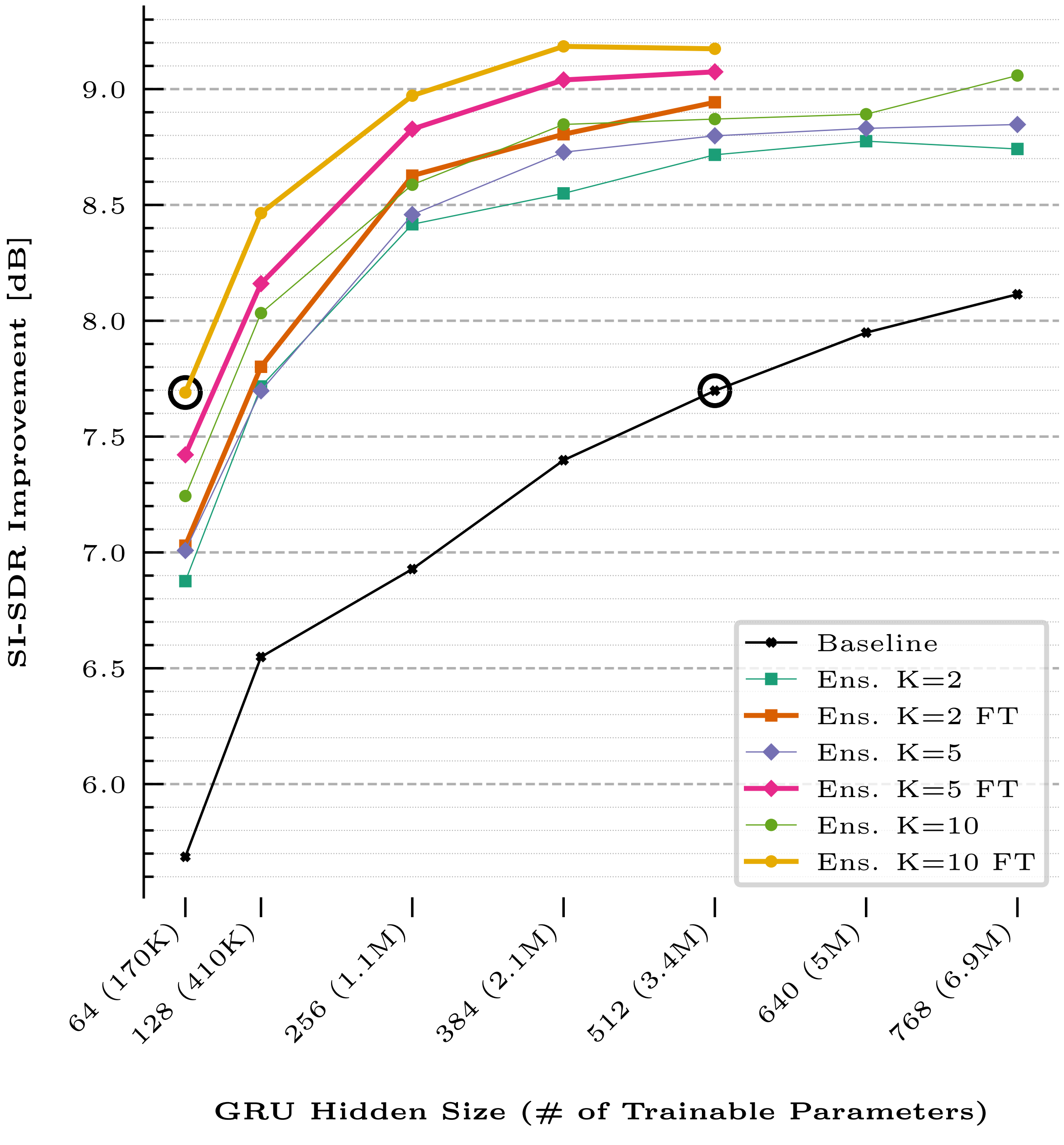
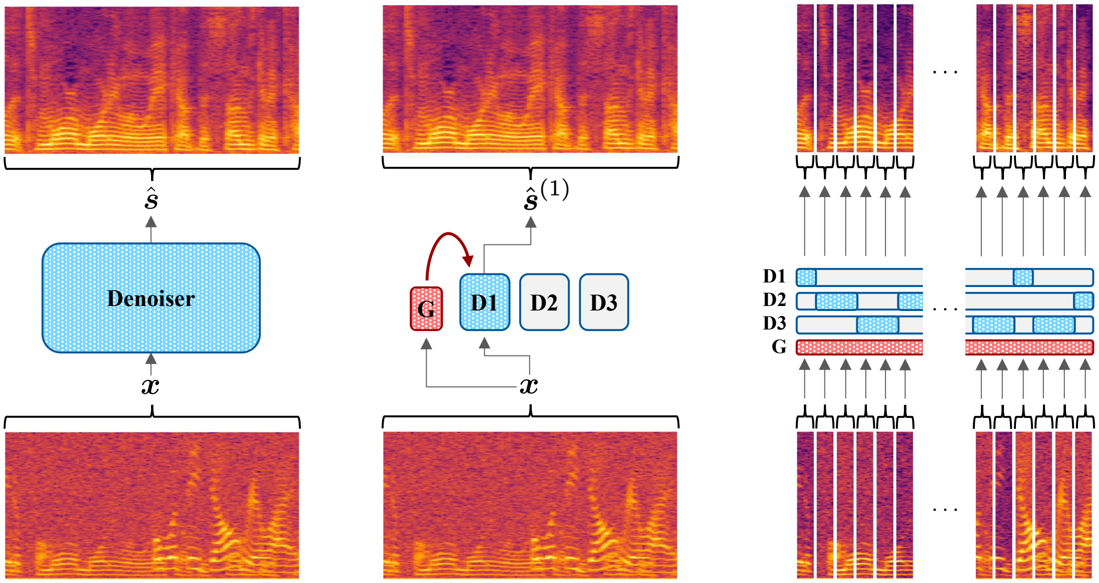


Figure 4.4: Comparison of speech enhancement performance between a baseline general-purpose model against different configurations of speaker-informed sparse ensemble models.

4.3 Block-Sparse RNNs for Universal Speech Modeling

A few questions arise regarding the limitations to the previously discussed sparse ensemble of specialists. Firstly, in our earlier experiments, the gating module selected the best-case specialist on a per-utterance basis. For online or streaming applications, the idea of an isolated utterance is ill-defined—therefore it may be better to have the model decide to switch experts on a rolling basis. To attain real-time performance, the obvious approach would be to reduce the lookback (input buffer) of the ensemble model. Secondly, our model grouping strategy (by subdividing the SE problem space) is a very hand-crafted procedure. We had to define semantically meaningful latent spaces (e.g., input SNR, speaker gender, etc.) where each specialist would focus on a non-overlapping subset of input cases; it may be possible that there is a non-semantic grouping of input cases which could yield even further improved performance. Lastly, because the hand-crafted grouping strategies may be sub-optimal, it is possible that there are redundancies between the specialists.

We hypothesize that a less-exclusive more-optimal grouping strategy may be possible. Additionally, rather than having a single specialist process an entire input sequence, it may be more adaptive and performant to quickly switch between specialists. Furthermore, it may be possible for a model to learn its own grouping strategy based solely on the acoustics instead of semantics. In this section, we introduce a modified recurrent neural network (RNN) which extends the idea of “adaptation by model selection” to do online real-time processing. With Fig. 4.5, we illustrate how the proposed block-sparse gated recurrent unit (BSGRU) may be viewed as a real-time extension of the previously discussed sparse ensemble of specialists from Section 4.1.



(a) (b) (c)

Figure 4.5: Visual comparison between (a) the non-adaptive generalist SE model, (b) the sparse ensemble of specialists model introduced in Section 4.1, and (c) the BSGRU introduced in this section. Only the BSGRU applies the "model selection" paradigm on a frame-by-frame basis to achieve real-time adaptation. Parameters with filled in colors indicate their usage during inference—in other words, the generalist uses all available model parameters, the sparse ensemble uses only the gating module and a single specialist, and the BSGRU switches between specialists over time with an always-active gating module.

4.3.1 Design

Conventional gated recurrent unit (GRU) The GRU was proposed as a more efficient easier-to-implement alternative to the long short-term memory (LSTM) unit for recurrent neural networks [110]. It processes sequential input to produce a hidden state by selectively retaining or forgetting information over time thanks to two gating mechanisms. The hidden state at time step t is computed as follows:

For our discussion, d denotes the number of input features and e is the number of output features, $\boldsymbol{x}_t \in \mathbb{R}^d$ is the input vector, $\boldsymbol{h}_t \in \mathbb{R}^e$ is the output vector, $\boldsymbol{r}_t \in (0,1)^e$ is the reset gate vector, $\boldsymbol{z}_t \in (0,1)^e$ is the update gate vector, $\hat{\boldsymbol{h}}_t \in \mathbb{R}^e$ is the candidate vector, and $\odot$ is the Hadamard product. There are a total of six matrices that comprise the model parameters.

```
Algorithm 1: GRU feed-forward                                   $\mathbf{A}(\boldsymbol{x}, \boldsymbol{h}_{t-1}; \boldsymbol{W}, \boldsymbol{U})$
  Input: $\boldsymbol{x}_t, \boldsymbol{h}_{t-1}$
  Initialize: $\boldsymbol{z}_t, \boldsymbol{r}_t, \hat{\boldsymbol{h}}_t, \boldsymbol{h}_t \leftarrow (\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0})$
  $\boldsymbol{r}_t \leftarrow \text{Sigmoid}\left(\boldsymbol{W}_r \boldsymbol{x}_t + \boldsymbol{U}_r \boldsymbol{h}_{t-1}\right)$                    // compute reset gate
  $\boldsymbol{z}_t \leftarrow \text{Sigmoid}\left(\boldsymbol{W}_z \boldsymbol{x}_t + \boldsymbol{U}_z \boldsymbol{h}_{t-1}\right)$                    // compute update gate
  $\hat{\boldsymbol{h}}_t \leftarrow \text{Tanh}\left(\boldsymbol{W}_h \boldsymbol{x}_t + \boldsymbol{U}_h \left[\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1}\right]\right)$         // compute candidate
  $\boldsymbol{h}_t \leftarrow \boldsymbol{z}_t \odot \hat{\boldsymbol{h}}_t + (1 - \boldsymbol{z}_t) \odot \boldsymbol{h}_{t-1}$
  Output: $\boldsymbol{h}_t$
```

These are conventionally denoted as matrices $\boldsymbol{W}$ and $\boldsymbol{U}$ representing the input-to-hidden and hidden-to-hidden mappings, respectively. The matrices are indexed with three subscripts $(r/z/h)$ to the specific computing the reset gate, update gate, or candidate vector. The gates control the flow of information—effectively, a recurrent unit which captures short-term memory will have a highly active reset gate, whereas one capturing long-term memory will have a highly active update gate. Bias terms are omitted from Alg. 1 for brevity. We denote $\mathbf{A}$ to be the mapping function for the conventional GRU which follows Alg. 1—i.e., $\boldsymbol{h}_t = \mathbf{A}(\boldsymbol{x}, \boldsymbol{h}_{t-1}; \boldsymbol{W}, \boldsymbol{U})$.

In this configuration, the GRU utilizes its entire parameter space to transform the input, doing so without explicitly modeling non-stationary groupings. Similar to most real-world sequential data, speech signals can also be modeled as sampling through discrete latent time-varying groups (e.g., segmental SNR, phonemes, vocal inflections, etc.). In order to motivate the recurrent network to learn discrete groupings within the data, we reformulate the GRU such that the parameters may be subdivided into "blocks".

The derivation for our proposed block-sparse gated recurrent unit (BSGRU) is as follows: first, we denote M to be the number of blocks. Then the number of hidden units per block is $b = \lfloor \frac{e}{M} \rfloor$ for hidden size e. Next, we reframe the model parameters $\boldsymbol{W}$ and $\boldsymbol{U}$ as block matrices which may be indexed (notated with a superscript). Similarly, the gate and candidate vectors can also be subdivided into "block vectors", e.g.,

$$\boldsymbol{r}_t = \begin{bmatrix} \boldsymbol{r}_t^{(1)} \\ \boldsymbol{r}_t^{(2)} \\ \vdots \\ \boldsymbol{r}_t^{(M)} \end{bmatrix} \quad \boldsymbol{z}_t = \begin{bmatrix} \boldsymbol{z}_t^{(1)} \\ \boldsymbol{z}_t^{(2)} \\ \vdots \\ \boldsymbol{z}_t^{(M)} \end{bmatrix} \quad \boldsymbol{W}_z = \begin{bmatrix} \boldsymbol{W}_z^{(1)} \\ \boldsymbol{W}_z^{(2)} \\ \vdots \\ \boldsymbol{W}_z^{(M)} \end{bmatrix} \quad \boldsymbol{U}_z = \begin{bmatrix} \boldsymbol{U}_z^{(1,1)} & \boldsymbol{U}_z^{(1,2)} & \cdots & \boldsymbol{U}_z^{(1,M)} \\ \boldsymbol{U}_z^{(2,1)} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \boldsymbol{U}_z^{(M,1)} & \cdots & \cdots & \boldsymbol{U}_z^{(M,M)} \end{bmatrix} \tag{4.4}$$

BSGRU: Gating Sub-Unit At each time step t, we estimate a belief vector $\boldsymbol{k}_t \in (0,1)^M$ whose maximum indicates the current "block index". This estimation is done using a "gating sub-unit" within the BSGRU. Similar to a Markov process, we design the current belief vector $\boldsymbol{k}_t$ to only be dependent on the current input $\boldsymbol{x}_t$ and the previous belief vector $\boldsymbol{k}_{t-1}$. In order to motivate the model to select only one group per input, we make $\boldsymbol{k}_t$ more sparse using a saturated softmax with temperature parameter λ. This temperature scalar controls the entropy of the softmax distribution, while preserving the relative ranks of each element. During inference (i.e., "evaluation mode"), the saturated softmax is replaced with a hard decision; this is theoretically equivalent to setting the temperature $\lambda \to \infty$. We describe the feed-forward operation of the gating sub-unit below:

The gating sub-unit is responsible for determining the current block index with respect to the current input and the previous block index. Our aim is for this classification task to incur the smallest possible computational overhead, therefore the hidden size of the gating sub-unit should be smaller than the specialist sub-unit hidden size, i.e., $e^* \leq e$.

In Alg. 2, we denote $\boldsymbol{r}^* \in (0,1)^{e^*}$ and $\boldsymbol{z}^* \in (0,1)^{e^*}$ to be the reset and update gate vectors for the gating sub-unit. The current hidden state is $\boldsymbol{h}_t^* \in \mathbb{R}^{e^*}$. We use the asterisk superscript to indicate gating sub-unit intermediate outputs. The trainable parameters of the gating sub-unit are $\boldsymbol{V} \in \mathbb{R}^{e^* \times d}$, $\boldsymbol{T} \in \mathbb{R}^{e^* \times e^*}$, and $\boldsymbol{Q} \in \mathbb{R}^{M \times e^*}$. The first two matrices affect the current input and prior hidden state, respectively. Notably, matrix $\boldsymbol{Q}$ enables the linear transformation from the gating sub-unit's hidden state to a belief logit vector

Algorithm 2: BSGRU gating sub-unit feed-forward $\mathbf{G}(\boldsymbol{x}, \boldsymbol{h}^*_{t-1}, \lambda; \boldsymbol{V}, \boldsymbol{T}, \boldsymbol{Q})$

```
Input: $\boldsymbol{x}_t, \boldsymbol{h}^*_{t-1}, \lambda$
Initialize: $\boldsymbol{k}_t \leftarrow \mathbf{0}$
$\boldsymbol{r}^*_t \leftarrow \text{Sigmoid}\left(\boldsymbol{V}_r \boldsymbol{x}_t + \boldsymbol{T}_r \boldsymbol{h}^*_{t-1}\right)$
$\boldsymbol{z}^*_t \leftarrow \text{Sigmoid}\left(\boldsymbol{V}_z \boldsymbol{x}_t + \boldsymbol{T}_z \boldsymbol{h}^*_{t-1}\right)$
$\hat{\boldsymbol{h}}^*_t \leftarrow \text{Tanh}\left(\boldsymbol{V}_h \boldsymbol{x}_t + \boldsymbol{T}_h \left[\boldsymbol{r}^*_t \odot \boldsymbol{h}^*_{t-1}\right]\right)$
$\boldsymbol{h}^*_t \leftarrow \boldsymbol{z}^*_t \odot \boldsymbol{h}^*_{t-1} + (1 - \boldsymbol{z}^*_t) \odot \hat{\boldsymbol{h}}^*_t$
$\hat{\boldsymbol{k}}_t \leftarrow \boldsymbol{Q}\boldsymbol{h}^*_t$          // linear map from hidden-to-belief vector space
if $stage = \texttt{test}$ then
    $i^* \leftarrow \arg\max_{0 \le i \le M} \hat{k}^{(i)}_t$          // get current block index
    $k^{(i^*)}_t \leftarrow 1$          // make belief vector one-hot (non-differentiable)
else
    $\boldsymbol{k}_t \leftarrow \text{Softmax}(\lambda \cdot \hat{\boldsymbol{k}}_t)$          // use saturated softmax (differentiable)
Output: $\boldsymbol{k}_t, \boldsymbol{h}^*_t$
```

space—i.e., the mapping $\mathbb{R}^{e^*} \to \mathbb{R}^M$. Lastly, the belief logit vector $\hat{\boldsymbol{k}}_t$ is converted to a probability vector $\boldsymbol{k}_t \in (0,1)^M$ using either a hard-max or softmax (the latter used only during training to produce a valid gradient). We represent the gating sub-unit of the BSGRU as $\mathbf{G}$ following Alg. 2—i.e., $\boldsymbol{k}_t = \mathbf{G}(\boldsymbol{x}, \boldsymbol{h}^*_{t-1}, \lambda; \boldsymbol{V}, \boldsymbol{T}, \boldsymbol{Q})$.

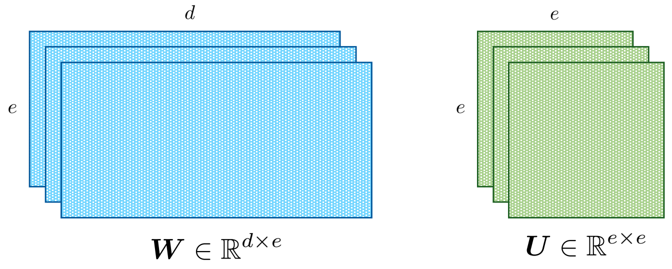
BSGRU: Specialist Sub-Unit Finally, the belief vector $\boldsymbol{k}_t$ is used to sparsely activate only a portion of the weight matrices. Note that the belief vector is a binary vector with M elements, and that the BSGRU specialist parameters can be subdivided into M-separate block vectors and matrices as shown in Eq. (4.4). During training time, multiplying each element of $\boldsymbol{k}_t$ to each block (from 1 to M) enables the sparse computation. At evaluation time, we use $\arg\max$ to select only the specialist model weights corresponding to a single block index. We incorporate dependence on the previous belief vector $\boldsymbol{k}_{t-1}$ to the specialist computation, allowing the model to transition between block states.

Algorithm 3: BSGRU specialist sub-unit feed-forward

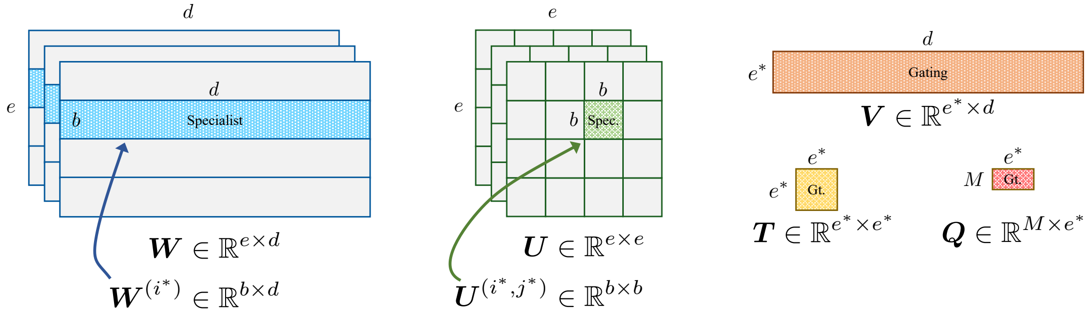
```
Input: $\boldsymbol{x}_t, \boldsymbol{h}_{t-1}, \boldsymbol{k}_t, \boldsymbol{k}_{t-1}$
Initialize: $\boldsymbol{z}_t, \boldsymbol{r}_t, \hat{\boldsymbol{h}}_t, \boldsymbol{h}_t \leftarrow (\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0})$
if evaluation mode then
    $i^* \leftarrow \arg\max_{1 \le i \le M} k_t^{(i)}$    // get current block index
    $j^* \leftarrow \arg\max_{1 \le j \le M} k_{t-1}^{(j)}$    // get previous block index
    $\boldsymbol{r}_t^{(i^*)} \leftarrow \text{Sigmoid}\left(\boldsymbol{W}_r^{(i^*)} \boldsymbol{x}_t + \boldsymbol{U}_r^{(i^*,j^*)} \boldsymbol{h}_{t-1}^{(j^*)}\right)$    // compute sparse reset gate
    $\boldsymbol{z}_t^{(i^*)} \leftarrow \text{Sigmoid}\left(\boldsymbol{W}_z^{(i^*)} \boldsymbol{x}_t + \boldsymbol{U}_z^{(i^*,j^*)} \boldsymbol{h}_{t-1}^{(j^*)}\right)$    // compute sparse update gate
    $\hat{\boldsymbol{h}}_t^{(i^*)} \leftarrow \text{Tanh}\left(\boldsymbol{W}_h^{(i^*)} \boldsymbol{x}_t + \boldsymbol{U}_h^{(i^*,j^*)} \left[\boldsymbol{r}_t^{(i^*)} \odot \boldsymbol{h}_{t-1}^{(j^*)}\right]\right)$    // compute sparse candidate
    $\boldsymbol{h}_t^{(i^*)} \leftarrow \boldsymbol{z}_t^{(i^*)} \odot \hat{\boldsymbol{h}}_t^{(i^*)} + \left(1 - \boldsymbol{z}_t^{(i^*)}\right) \odot \boldsymbol{h}_{t-1}^{(j^*)}$
else if training mode then
    for $i \leftarrow 1$ to $M$ do    // iterate over all block indices
        $\boldsymbol{r}_t^{(i)} \leftarrow \text{Sigmoid}\left(k_t^{(i)} \boldsymbol{W}_r^{(i)} \boldsymbol{x}_t + \sum_{j=0}^{M} \left[k_t^{(i)} k_{t-1}^{(j)} \boldsymbol{U}_r^{(i,j)} \boldsymbol{h}_{t-1}^{(j)}\right]\right)$
        $\boldsymbol{z}_t^{(i)} \leftarrow \text{Sigmoid}\left(k_t^{(i)} \boldsymbol{W}_z^{(i)} \boldsymbol{x}_t + \sum_{j=0}^{M} \left[k_t^{(i)} k_{t-1}^{(j)} \boldsymbol{U}_z^{(i,j)} \boldsymbol{h}_{t-1}^{(j)}\right]\right)$
        $\hat{\boldsymbol{h}}_t^{(i)} \leftarrow \text{Tanh}\left(k_t^{(i)} \boldsymbol{W}_h^{(i)} \boldsymbol{x}_t + \sum_{j=0}^{M} \left[k_t^{(i)} k_{t-1}^{(j)} \boldsymbol{U}_h^{(i,j)} \left(\boldsymbol{r}_t^{(i)} \odot \boldsymbol{h}_{t-1}^{(j)}\right)\right]\right)$
        $\boldsymbol{h}_t^{(i)} \leftarrow \boldsymbol{z}_t^{(i)} \odot \left[k_t^{(i)} \hat{\boldsymbol{h}}_t^{(i)}\right] + \left(1 - \boldsymbol{z}_t^{(i)}\right) \odot \sum_{j=0}^{M} \left[k_{t-1}^{(j)} \boldsymbol{h}_{t-1}^{(j)}\right]$
Output: $\boldsymbol{h}_t$
```

The current and previous outputs of the gating sub-unit ($\boldsymbol{k}_t$ and $\boldsymbol{k}_{t-1}$) are forwarded to a specialist sub-unit in order to sparsely activate its weights. In this way, the specialist sub-unit learns an adaptive SE function. We define the $\arg\max$ of $\boldsymbol{k}_t$ as the current block index i; subsequently the $\arg\max$ of $\boldsymbol{k}_{t-1}$ as the prior block index j. Unlike the conventional GRU which uses all of its parameters $\boldsymbol{W}$ and $\boldsymbol{U}$, in the BSGRU, the specialist sub-unit selects a specific block matrix within $\boldsymbol{W}$ and another specific block matrix within $\boldsymbol{U}$. We represent the specialist sub-unit of the BSGRU as $\mathbf{B}$ following Alg. 3—i.e., $\boldsymbol{h}_t = \mathbf{B}(\boldsymbol{x}_t, \boldsymbol{h}_{t-1}, \boldsymbol{k}_t, \boldsymbol{k}_{t-1}; \boldsymbol{W}, \boldsymbol{U})$.

Fig. 4.6 provides a visual comparison between the six weight matrices present in a conventional GRU versus the nine weight matrices in our proposed BSGRU. As shown in the figure, at inference-time / evaluation, only the parameters which are colored are used. Our proposed model achieves run-time complexity savings when the number of active parameters in the BSGRU are less than that of a fully-active GRU—the hyper-parameters that impact this are: the choice of hidden size (e), the gating sub-unit overhead (e^*), and the number of blocks (M). Fig. 4.7 shows the flow of the various input and output variables in a conventional GRU and in our proposed BSGRU.

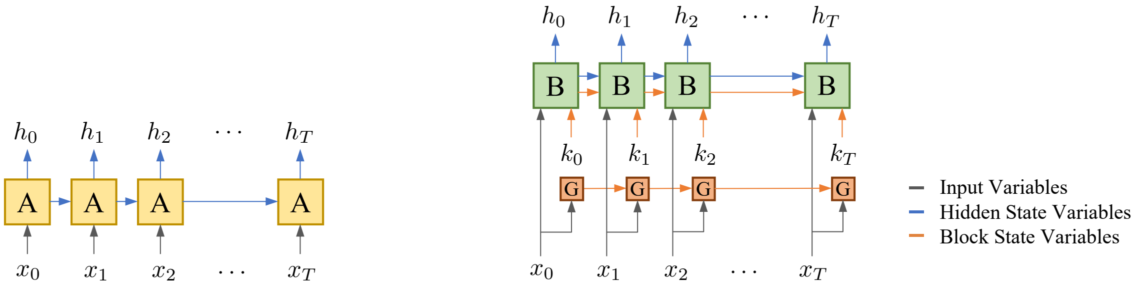

d
e
e
e
$\boldsymbol{W} \in \mathbb{R}^{d\times e}$
$\boldsymbol{U} \in \mathbb{R}^{e\times e}$

(a) Conventional GRU.


d
e
d
b
Specialist
$\boldsymbol{W} \in \mathbb{R}^{e\times d}$
$\boldsymbol{W}^{(i^*)} \in \mathbb{R}^{b\times d}$
e
e
b
b
Spec.
$\boldsymbol{U} \in \mathbb{R}^{e\times e}$
$\boldsymbol{U}^{(i^*,j^*)} \in \mathbb{R}^{b\times b}$
d
e^*
Gating
$\boldsymbol{V} \in \mathbb{R}^{e^*\times d}$
e^*
e^*
Gt.
$\boldsymbol{T} \in \mathbb{R}^{e^*\times e^*}$
e^*
M
Gt.
$\boldsymbol{Q} \in \mathbb{R}^{M\times e^*}$

(b) Proposed BSGRU.

Figure 4.6: Comparison of the weight matrices. As the BSGRU weights are divided into M blocks, the number of hidden units per block is $b = \lfloor \frac{e}{M} \rfloor$. In a GRU feed-forward computation, the entirety of the model parameters are utilized and updated, whereas with the BSGRU, only a subset of the model parameters are used. This subset is determined by i^* and j^* which are the block indexes at time t and $t-1$. Specifically in this example, $M = 4$, $i^* = 2$, and $j^* = 3$, indicating an inter-block transition. Accounting for the gates, the conventional GRU has six weight matrices, whereas the BSGRU has nine weight matrices, six of which are sparsely active.



(a) Conventional GRU.

(b) Proposed BSGRU.

Figure 4.7: An "unrolled" representation of the conventional GRU shows that the recurrent unit **A** computes the current hidden state $\boldsymbol{h}_t$ using only the current input $\boldsymbol{x}_t$ along with the previous hidden state $\boldsymbol{h}_{t-1}$ (as described in Alg. 1). Our proposed BSGRU is effectively two sub-units—the gating sub-unit **G** first computes a block state $\boldsymbol{k}_t$ using the current input and the previous block state $\boldsymbol{k}_{t-1}$. Next, the specialists sub-unit **B** harmonizes the current and previous block states along with the current input to estimate the current hidden state. **G** and **B** follow Alg. 2 and Alg. 3, respectively.

4.3.2 Pretraining Process

Training a BSGRU is a complex optimization task which has many potentials for failure similar to a generative adversarial network (GAN); this is because the BSGRU jointly classifies and regresses over the same input signal. Although the BSGRU can theoretically learn latent groupings in the sequential data, we find that empirically some pre-training is required in order to prevent the model from collapsing to a sub-optimal solution—for example, if the BSGRU learns to use only one or a few blocks out of all M possible blocks.

We prevent this collapse by formulating a two-stage pre-training procedure: first we optimize the specialist sub-unit **B** and then the gating sub-unit **G**, in that order. For pre-training, a latent space which can divided into M discrete groups (e.g., speaker characteristics, SNR levels) must be chosen. Then, the ground-truth group label (i.e., block index) for the training data input at any time t must be known—for example, if the latent space is "segmental SNR levels", then the group label for a single frame of the noisy input spectrogram $\boldsymbol{X}_t$ is simply the binned value of the the segmental SNR for that frame. We discuss the exact binning procedure used for our experiments later in Section 4.3.3.

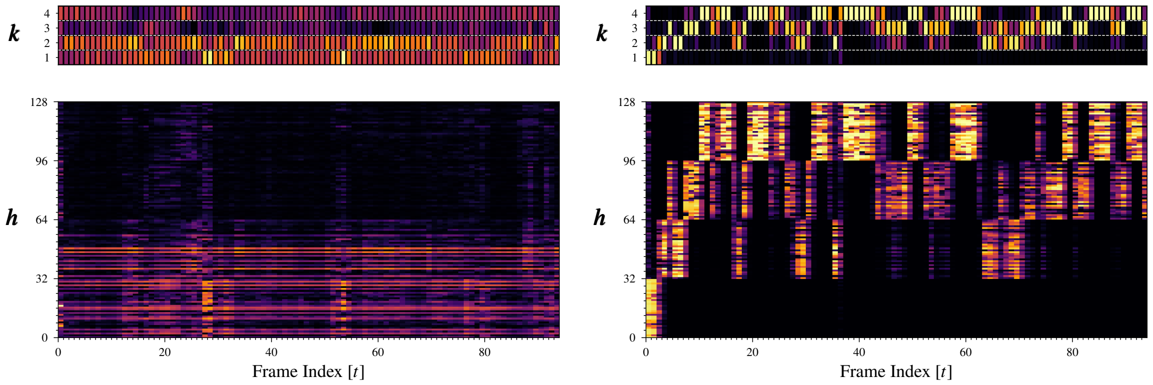
For the first stage, each of the blocks within the specialist sub-unit parameters ($\boldsymbol{W}^{(i)}$ and $\boldsymbol{U}^{(i,j)}$ for $i, j \in \{1, \ldots, M\}$) are sparsely activated and individually adapted following Alg. 3. This can be done by swapping the untrained classifier (the gating sub-unit $\mathbf{G}$) with a hypothetical *oracle classifier* that outputs only the ground-truth block index any input at all times t. Practically, this is equivalent to substituting all uses of the belief vector $\boldsymbol{k}$ in Alg. 3 with the one-hot representation of the ground-truth block index $\mathring{\boldsymbol{k}}$. Then, at any time t, $i^* = \underset{1 \le i \le M}{\arg\max}\, \mathring{\boldsymbol{k}}_t^{(i)}$ and $j^* = \underset{1 \le j \le M}{\arg\max}\, \mathring{\boldsymbol{k}}_{t-1}^{(j)}$—because i^* and j^* are the ground-truth bin indices, only the best-suited specialist block matrices ($\boldsymbol{W}^{(i^*)}$ and $\boldsymbol{U}^{(i^*,j^*)}$) will be used/updated.

In the second stage, all of the parameters of $\mathbf{B}$ are kept fixed. We train only the gating sub-unit parameters ($\boldsymbol{T}$, $\boldsymbol{V}$, and $\boldsymbol{Q}$). Although the gating sub-unit's task is to classify the input, the optimization criterion is still to minimize the discrepancy between the overall BSGRU output and the expected output ($\mathcal{E}(\boldsymbol{M}, \hat{\boldsymbol{M}})$)—in other words, the training loss is based on regression (e.g., MSE) and not classification (e.g., cross-entropy (CE)).

After this two-stage pre-training, it is still possible to fine-tune the full BSGRU over the training corpus. This allows the specialist sub-unit to adjust its parameters to account for the gating sub-unit's misclassified inputs. Fig. 4.8 shows the hidden state output vector $\boldsymbol{h}$ and the belief state vector $\boldsymbol{k}$ over all time t for a BSGRU which has completed both stages of pre-training.

4.3.3 Experiment Setup

Architecture We devise an experiment using a BSGRU for adaptive online speech enhancement. As explained in Section 2.3, we develop TF-masking models which take noisy input speech spectrogram $\boldsymbol{X}$ and estimate a binary mask $\boldsymbol{M}$ as output. All spectrograms are computed using the STFT with 1024-point Hann windows and 75 % overlap—the resulting shape of matrices $\boldsymbol{X}$ and $\boldsymbol{M}$ is $513 \times L$, where L is the number of frames. Although more

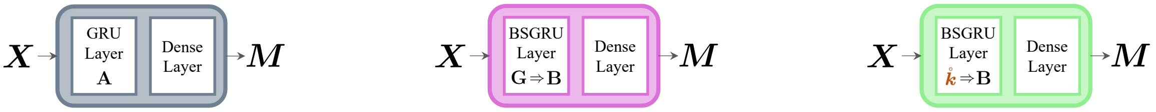


(a) without pre-training (b) after 5 epochs

Figure 4.8: BSGRU belief vector $\boldsymbol{k}$ and hidden state output $\boldsymbol{h}$ observed during Stage 2 pre-training over various epochs. In this example, the BSGRU hidden size $e = 128$ and the number of blocks $M = 4$, making the "effective hidden size" $b = \lfloor \frac{e}{M} \rfloor = 32$. Without pre-training, the gating sub-unit makes arbitrary choices for the block index, so $\boldsymbol{k}$ is initially non-sparse and only the first or second specialist pass-through the input signal. With more training epochs, the gating sub-unit generates a sparser belief vector, thereby enforcing a sparser hidden vector.

complex models for SE are practical, we limit the tested architectures to use only a single recurrent network layer in order to do a targeted ablation.

As the goal is to assess the specific contributions of our proposed block-sparsity—the compared models are equivalent except that the recurrent layer is either a standard GRU (baseline) or BSGRU (proposed). The input dimensionality is the number of frequency bins ($1 + \lfloor \frac{1024}{2} \rfloor = 513$), and the dimensionality of the recurrent layer output is the hidden size (e). The hidden state is then input to a trainable dense layer, which maps back to the input dimension 513. We survey three choices of "effective recurrent layer hidden size" in order to observe the effect of model size on the benefits of adaptation. Our choices are: $(e_{\text{GRU}}, b_{\text{BSGRU}}) = 32$, 128, or 512. We use the term "effective" to indicate that the BSGRU specialist sub-unit only uses a fraction of its hidden size per time step, whereas the standard GRU leverages its full hidden size. For fair comparison, we set $e_{\text{GRU}} = b_{\text{BSGRU}}$, because $e_{\text{BSGRU}} = b_{\text{BSGRU}} \cdot M$.



(a) GRU Baseline (b) Pre-Trained BSGRU (c) Oracle BSGRU

Figure 4.9: Architectural differences between experiment models. In the oracle model, in place of computing the belief state vector $\boldsymbol{k}$ using the gating sub-unit $\mathbf{G}$, we instead assume hypothetical access to the ground-truth block index $\mathring{\boldsymbol{k}}$ as defined in Section 4.3.3.

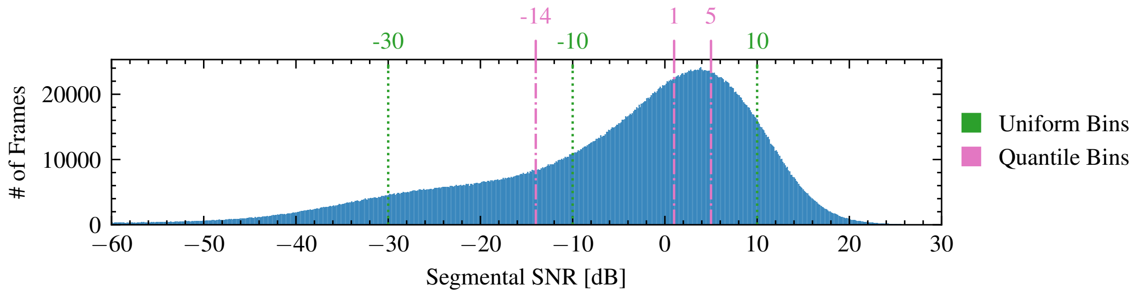


Figure 4.10: Distribution of the segmental SNR values present in the LibriMix single-speaker enhancement training corpus.

Latent Space We show a BSGRU with $M = 4$ blocks adapting to test-time noisy speech based on "segmental SNR level". Because segmental SNR is a continuous value and we have $M = 4$ groups, we consider two possible binning strategies, accounting for the distribution of the LibriMix single-speaker enhancement training corpus segmental SNR as shown in Fig. 4.10. The first naïve binning strategy would be to use evenly spaced intervals which span the breadth of all possible segmental SNR values. We refer to this as the "uniform binning" approach, and as $M = 4$, we arbitrarily define three bin edges: $(-30, -10, 10)$ dB. Subsequently, the ground-truth block index encoded as a one-hot vector $(\mathring{\boldsymbol{k}}_t)$ is defined as follows:

$$\mathring{\boldsymbol{k}}_t^{\text{Uniform}} = \begin{cases} [1\ 0\ 0\ 0] & \text{SegSNR}(\boldsymbol{X}_t) \leq -30\,\text{dB} \\ [0\ 1\ 0\ 0] & -30\,\text{dB} < \text{SegSNR}(\boldsymbol{X}_t) \leq -10\,\text{dB} \\ [0\ 0\ 1\ 0] & -10\,\text{dB} < \text{SegSNR}(\boldsymbol{X}_t) \leq 10\,\text{dB} \\ [0\ 0\ 0\ 1] & 10\,\text{dB} < \text{SegSNR}(\boldsymbol{X}_t) \end{cases} \tag{4.5}$$

A more analytical second binning strategy could be based on quantiles. This maximizes the likelihood that each specialist within **B** sees the same number of input frames. In the "quantile binning" approach, as $M = 4$, the bin edges are empirically derived at 25%, 50%, and 75% splits: $(-14, -1, 5)$ dB. Similarly, the ground-truth block index encoded as a one-hot vector is defined as follows:

$$\mathring{\boldsymbol{k}}_t^{\text{Quantile}} = \begin{cases} [1\ 0\ 0\ 0] & \text{SegSNR}(\boldsymbol{X}_t) \leq -14\,\text{dB} \\ [0\ 1\ 0\ 0] & -14\,\text{dB} < \text{SegSNR}(\boldsymbol{X}_t) \leq -1\,\text{dB} \\ [0\ 0\ 1\ 0] & -1\,\text{dB} < \text{SegSNR}(\boldsymbol{X}_t) \leq 5\,\text{dB} \\ [0\ 0\ 0\ 1] & 5\,\text{dB} < \text{SegSNR}(\boldsymbol{X}_t) \end{cases} \tag{4.6}$$

The exact bin edges are annotated in Fig. 4.10; these bin edge values are specific to our choice of using the "segmental SNR" latent space. Choosing another latent space would necessitate re-adjusting the bin edges. In our experiments, we assess the effect of these two strategies (uniform vs. quantile) with respect to the denoising performance of a single-layer BSGRU.

Dataset This experiment utilizes the LibriMix [36] dataset; it is an open-source recipe for combining clean speech recordings from "Librispeech" [30] with ambient noise recordings from "WHAM!" [111] to produce a deterministic set of mixture audio. In particular, we use only the mixture audio containing a single speaker contaminated by background noise; the dataset additionally supports two- and three-speaker mixtures. All derived audio files are sampled at 16 kHz, but inputs to the model are truncated to be 3 s in duration. In total, there is 58 h of training data, 11 h of validation data, and 11 h of test data. Speaker identities are non-overlapping between the three partitions.

Hyperparameters For all trials, we use $M = 4$ for the number of blocks. With our BSGRUs, we fix the gating sub-unit hidden size $e^* = 16$; this results in a very small computational overhead for computing Alg. 2. We find that performance is largely unaffected by different values for the gating sub-unit temperature $\lambda = 1$, $1\mathrm{e}-1$, and $1\mathrm{e}-2$. The Adam optimizer [95] is used for all trials with various choices of learning rates $\eta = 1\mathrm{e}-3$, $1\mathrm{e}-4$ or $1\mathrm{e}-5$; reported performance for each model is the best-performing. Also, our training loss function is the negative SISDR between the estimated clean speech $\hat{\boldsymbol{y}}$ and the ground-truth clean speech $\boldsymbol{M}$. Recall that $\hat{\boldsymbol{y}} = \boldsymbol{M} \odot \boldsymbol{X}$, where $\boldsymbol{X}$ is the noisy speech input spectrogram and $\boldsymbol{M}$ is the TF binary ratio mask estimated by the denoiser model. We use a fixed choice of 100 epochs for both the baseline GRU and oracle BSGRU. The oracle BSGRU is used as the Stage 1 initialization ($\mathbf{B}$) for the pre-trained BSGRU, which goes through another 100 epochs to optimize only $\mathbf{G}$.

4.3.4 Results

Fig. 4.11 summarizes the results of the experiment. Reported SISDR improvement values are the averages ± 95 %-confidence interval. Note that the binning strategy axis only applies to the BSGRU, so the baseline GRU numbers are equivalent in both rows.

Oracle Binning Strategy Firstly, across all configurations, we see that the oracle BSGRU model achieves the most significant boost in performance over the the non-adapted GRU model. This is to be expected, as the oracle model simulates a BSGRU with a perfect 100%-accurate gating sub-unit. Comparing uniform vs. quantile binning strategy, we see that the oracle quantile models perform best; this can be explained due to the fact that the quantile binning strategy, by definition, maximizes the utilization of all M specialist blocks to cover the near-equal quadrants of the "segmental SNR" latent space. In other words, the number of most-suited input cases is well-balanced among the M blocks. In the

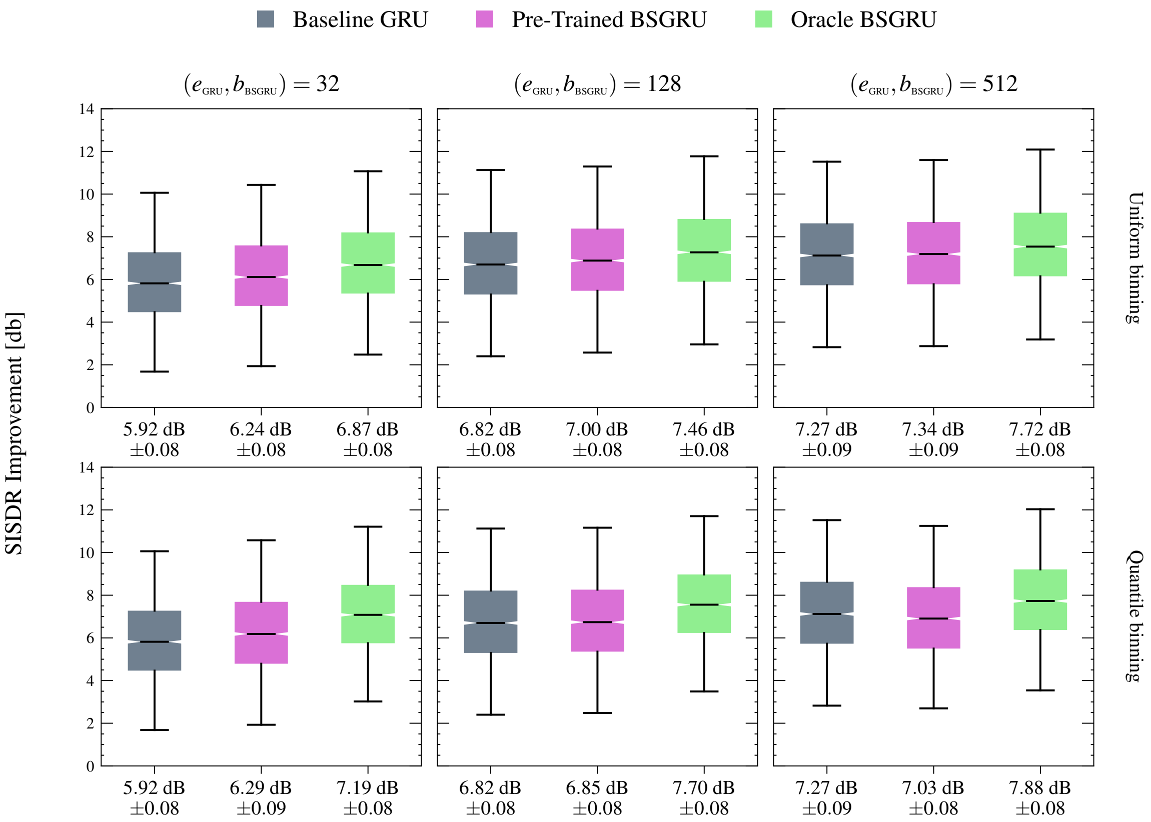


Figure 4.11: Comparison of speech enhancement performance between a unadapted GRU baseline model, the pre-trained proposed BSGRU, and an oracle BSGRU.

uniform binning approach, the 3$^{\text{rd}}$ specialist would encounter the vast majority of input frames, whereas the 1$^{\text{st}}$ specialist would encounter the least; this relates to the area under the curve for each bin edge in Fig. 4.10. In summary, with uniform binning, the input cases are not well-balanced among the specialists, making it understandably subpar to the quantile binning.

Non-Oracle Binning Strategy Consequently, with the fully pre-trained BSGRU, the uniform binning strategy outperforms the quantile binning strategy over all model sizes. This may be explained by the fact that the gating sub-unit $\mathbf{G}$ is not perfect; even after the two-stage pre-training, accuracy may be at best 70 % to 80 %. Naturally, inputs may be misclassified; the uniform binning ensures that the centroids of each specialist are maximally spread out in the latent space. The bin edges are very close in value with quantile binning (e.g., 1 dB to 5 dB) as compared to uniform binning (e.g., −10 dB to 10 dB), therefore misclassification is more likely in the former. Thus, for practical applications, where the ground-truth block index $\mathring{\boldsymbol{k}}$ is not known at either training or test-time, a simple uniform binning of the latent space may be more performant.

Model Size Next, we see that for the smallest model size, the two-stage pre-trained BSGRU outperforms the non-adapted GRU model by a statistically significant amount (about 5.4 % to 6.3 % improvement). This improvement is shown for models with the same "effective hidden size", highlighting the merit of our proposed model that leverages its increased spatial complexity with negligible change in computational complexity. With the next largest model size, the performance gain from non-adapted to adapted is noticeably decreased (about 0.4 % to 2.6 % improvement). With the largest size, the performance is regressed with quantile binning and only negligibly improved with uniform binning (about −3.3 % to 0.9 % improvement). This finding echos our previous conclusions that the benefits of model adaptation are best realized with smaller model sizes. A smaller model stands to gain more from personalization given the reduced number of total parameters.

Chapter 5

Conclusion

The goal of this dissertation was to present two classes of algorithms that address the task of personalized (speaker-specific) speech enhancement, emphasizing the computational- and data-efficiency of these methods. Other research has shown that personalization expectedly improves performance for the target speaker, but the resource efficiency benefits discussed in this dissertation were previously unexplored.

A significant focus of our study was on lossless model compression. The core idea was to verify that personalized SE models could match or exceed the performance of non-personalized SE models using fewer model parameters. We saw that this was the case for specialist models pre-trained using NTT; for example, a tiny ConvTasNet pre-trained using contrastive mixtures matched the performance of a small ConvTasNet pre-trained using standard fully-supervised SE. Similarly, with our model selection experiments, we saw that a generalist GRU-based SE model using 384 hidden units was outperformed by a sparse ensemble of ten specialist GRU-based PSE models each using 64 hidden units. These examples show that the spatial complexity (i.e., the number of total stored parameters) could be reduced without degrading SE performance on the target speaker. Therefore, our assertion that personalization was a novel paradigm for lossless model compression was empirically validated.

Needless to say, using fewer model parameters minimizes both the space- and time-complexity of the PSE algorithms. In other words, by achieving lossless model compression, we have also inherently reduced the overall run-time (or inference) complexity. Particularly with the model selection / sparse ensemble paradigm, the savings on space- and time-complexity are, in fact, decoupled. For example, if storage space is not a limiting factor, one can simply increase K—the number of specialists in the ensemble—to improve the adapted or personalized performance even more (up to a saturating point). Despite increasing K, the actual algorithmic latency of the sparse ensemble remains the same because we select only one best-suited specialist model for inference. In other words, our fundamental exploitation of the MLE ensemble formulation enabled the opportunity to achieve isolated gains in run-time complexity.

Another priority of our work was to minimize or avoid any model pre-training using (reference quality) clean speech data from the target speaker. We defined our privacy goals in this regard because of the recent advancements in speech synthesis research, showing that realistic vocal forgery may be feasible with as little as 5 seconds of reference data. Our core hypothesis is that (non-reference quality) noisy speech data is unusable for training legitimate text-to-speech (TTS) systems, but may be beneficial for PSE through self-supervised learning. The proposed NTT algorithms (PseudoSE and contrastive mixtures (CM)) are pretext tasks meant to derive meaningful features distinctly for the PSE task. To this end, we made the assumption that noisy speech data was easier to collect—skipping the need for a voice enrollment process. Furthermore, we assumed that the SNR of the in-the-wild data followed a uniform distribution between 0 dB to 15 dB in accordance with prior literature in psychoacoustics. Naturally, our NTT methods may be limited depending on how originally degraded the in-the-wild data is; we therefore proposed data purification (DP) to minimize the delta between the pseudo- and real SE learning objectives. On the other end of the spectrum, our illustration of personalization through model selection assumes that target speaker data is wholly unavailable during training time, so we sidestep the privacy concern entirely.

However, if the target speaker's clean speech is accessible, all of our proposed PSE algorithms can benefit from additional transfer learning/fine-tuning. Although we only examined the success of fine-tuning in our NTT experiments, it is highly likely that the sparse ensembles or BSGRU models would also benefit from having any knowledge of the test-time speaker.

Furthermore, with the NTT experiments, we gleaned the impact of training SE models on in-domain data, even if it's noisy. Stemming from our experiment setup, we saw that multiple types of personalized models—leveraging only 25 minutes of referenceless noisy speech from the target speaker—were able to outperform a non-personalized generalist model—trained on 440 hours of reference-quality clean speech from 1000+ anonymous speakers. This massive reduction in training data size shows the potency of self-supervised learning with in-domain data versus fully-supervised learning over out-of-domain data. Subsequently, using a smaller training dataset reduces the overall elapsed training time, thereby reducing computing costs.

5.1 Contributions

We can summarize the novelty of this dissertation by viewing the proposed methods as broad frameworks for addressing PSE based on the availability of speaker-specific training data.

We affirm that "personalization via noisy-target training" is a robust approach when unlabeled noisy speaker data is available. Realistically, this noisy data is likely more abundant or, at least, easier to obtain. All of the self-supervised models could be improved further through fine-tuning when a small amount of the target speaker's clean speech data was available. As they were trained in-domain, the NTT models adapted more effectively than the out-of-domain fully-supervised models.

Next, we showed how "personalization via model selection" addresses the cases where no target speaker data is available. The number of specialist sub-modules K may be chosen based on the space limitations, enabling a variable subdivision of the overarching SE problem. The performance upper bound of model selection methods ultimately depends on the grouping strategy and the group transition mechanism.

5.2 Limitations & Future Work

One remaining claim from this dissertation's motivation was to address social fairness through model personalization. It would have been ideal to discover concrete evidence of our models achieving equivalent SE performance for under-recognized speakers, such as those with diverse accents, from different age groups, or even with specific speech disorders. This effort would have rendered further insights into the design of accessible speech-based machine learning systems. Regrettably, the absence of specific demographic annotations in many public speech datasets limited our ability to fully explore the potential of our methods. Although we could not investigate our claim in this dissertation, the methods discussed do not make any assumptions about specific speaker identities or characteristics. So, theoretically, they may be applied to the under-recognized cases, given the appropriate training data.

We note that additional studies could have been done regarding the data-efficiency arguments. For example, our NTT experiments were designed with the specialist models being trained on approximately 25 min of noisy speech from the target speaker. An extension of this work could have varied this amount, generating a curve to see the impact of in-domain noisy data on the pseudo SE learning objective. Also, because the DP method diminishes the learning contribution of overly degraded frames, it would have been informative to determine what percentage of the in-the-wild data was ultimately usable.

A few other supplementary experiments may have strengthened the arguments presented in this dissertation. For example, we could have assessed more sophisticated speech enhancement neural network architectures besides the described GRU-based masking network or ConvTasNet. There are a plethora of speech enhancement algorithms that may be classified as TF-masking or end-to-end signal estimation methods; our experimental validation only covers one algorithm from each approach. Also, more extensive sweeps over experiment hyperparameters—including learning rates, optimizers, and number of clusters (K)—may have been done with additional time. For our experiments, we opted for the simplest deformation function: a sum of speech and noise signals. However, we anticipate the proposed methods would translate well to more complex deformation functions that incorporate reverberation or other filters. Lastly, it would have been ideal to formulate an experiment that makes the NTT and model selection methods more directly comparable.

We recognize that many other studies about personalization directly identify the target speaker by estimating a "speaker ID" vector. However, the proposed methods of this dissertation are data-centric and intentionally do not involve an explicit speaker identification (SI) task. In that way, we empirically see that features learned for the SE task need not compromise the speaker's identity—that is, identification may not be essential for personalization. Therefore, we suspect that the availability of training data is likely the biggest factor in deciding the best framework for training and deploying a PSE system.

In short, we hope this dissertation inspires additional research on providing personalized experiences with speech-based systems, prioritizing resource efficiency and speaker privacy.

BIBLIOGRAPHY

[1] J. Chen and D. Wang. Long short-term memory for speaker generalization in supervised speech separation. *Journal of the Acoustical Society of America*, 141(6):4705–4714, 2017.

[2] S. Han, J. Pool, J. Tran, and W. Dally. Learning both Weights and Connections for Efficient Neural Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 28, pages 1135–1143, 2015.

[3] 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.

[4] E. Strubell, A. Ganesh, and A. McCallum. Energy and policy considerations for modern deep learning research. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(09):13693–13696, Apr. 2020.

[5] Steve Lohr. Facial Recognition Is Accurate, if You're a White Guy. *The New York Times*, Feb. 2019; Retrieved 2020-6-5.

[6] J. Buolamwini and Timnit G. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pages 77–91, 2018.

[7] 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*, pages 6462–6468. European Language Resources Association, 2020.

[8] A. Koenecke, A. Nam, E. Lake, J. Nudell, M. Quartey, Z. Mengesha, C. Toups, J. R. Rickford, D. Jurafsky, and S. Goel. Racial Disparities in Automated Speech Recognition. *Proc. of the National Academy of Sciences*, 117(14):7684–7689, 2020.

[9] 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*, 25(1):153–167, Jan 2017.

[10] O. Kimball, M. Schmidt, H. Gish, and J. Waterman. Speaker Verification with Limited Enrollment Data. In *Proc. Eurospeech*, pages 967–970, 1997.

[11] Man-Wai Mak, Roger Hsiao, and Brian Mak. A Comparison of Various Adaptation Methods for Speaker Verification With Limited Enrollment Data. In *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages 929–932, 2006.

[12] C. Zhang and K. Koishida. End-to-End Text-Independent Speaker Verification with Triplet Loss on Short Utterances. In *Proc. Interspeech*, 2017.

[13] J. Lau, B. Zimmerman, and F. Schaub. Alexa, Are You Listening? Privacy Perceptions, Concerns and Privacy-seeking Behaviors with Smart Speakers. *Proceedings of the ACM on Human-Computer Interaction (PACM HCI)*, 2(CSCW), 2018.

[14] J. Foehr and C. C. Germelmann. Alexa, Can I Trust You? Exploring Consumer Paths to Trust in Smart Voice-Interaction Technologies. *Journal of the Association for Consumer Research*, 5(2):181–205, 2020.

[15] M. Vimalkumar, S. K. Sharma, J. B. Singh, and Y. K. Dwivedi. ‘Okay Google, what about my privacy?’: User’s privacy perceptions and acceptance of voice based digital assistants. *Comput. Hum. Behav.*, 120(C), 2021.

[16] F. Acikgoz and R. P. Vega. The Role of Privacy Cynicism in Consumer Habits with Voice Assistants: A Technology Acceptance Model Perspective. *International Journal of Human–Computer Interaction*, 38(12):1138–1152, 2022.

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

[18] R. Tatman and C. Kasten. Effects of Talker Dialect, Gender & Race on Accuracy of Bing Speech and YouTube Automatic Captions. In *Proc. Interspeech*, pages 934–938, 2017.

[19] T. Hashimoto, M. Srivastava, H. Namkoong, and P. Liang. Fairness Without Demographics in Repeated Loss Minimization. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1929–1938, 2018.

[20] G. Chalhoub and I. Flechais. “Alexa, Are You Spying on Me?”: Exploring the Effect of User Experience on the Security and Privacy of Smart Speaker Users. In *HCI for Cybersecurity, Privacy and Trust*, pages 305–325. Springer International Publishing, 2020.

[21] A. Sivaraman and M. Kim. Sparse Mixture of Local Experts for Efficient Speech Enhancement. In *Proc. Interspeech*, pages 4526–4530, 2020.

[22] A. Sivaraman and M. Kim. Self-supervised learning for personalized speech enhancement. *arXiv preprint arXiv:2104.02017*, 2021.

[23] A. Sivaraman, S. Kim, and M. Kim. Personalized speech enhancement through self-supervised data augmentation and purification. In *Proc. Interspeech*, pages 2676–2680, 2021.

[24] 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.

[25] A. Sivaraman and M. Kim. Efficient Personalized Speech Enhancement Through Self-Supervised Learning. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1342–1356, 2022.

[26] E. C. Cherry. Some Experiments on the Recognition of Speech, with One and with Two Ears. *Journal of the Acoustical Society of America*, 25:975–979, 1953.

[27] J. H. McDermott. The cocktail party problem. *Current Biology*, 19(22):R1024–R1027, Dec 2009.

[28] S. Graetzer, J. Barker, T. J. Cox, M. Akeroyd, J. F. Culling, G. Naylor, E. Porter, and R. V. Muñoz. Clarity-2021 Challenges: Machine Learning Challenges for Advancing Hearing Aid Processing. In *Proc. Interspeech*, pages 686–690, 2021.

[29] H. Dubey, V. Gopal, R. Cutler, A. Aazami, S. Matusevych, S. Braun, S. E. Eskimez, M. Thakker, T. Yoshioka, H. Gamper, and R. Aichner. Icassp 2022 deep noise suppression challenge. In *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 9271–9275, 2022.

[30] 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)*, pages 5206–5210, 2015.

[31] C. Veaux, J. Yamagishi, and S. King. The Voice Bank Corpus: Design, Collection and Data Analysis of a Large Regional Accent Speech Database. In *2013 International Conference Oriental COCOSDA held jointly with 2013 Conference on Asian Spoken Language Research and Evaluation (O-COCOSDA/CASLRE)*, pages 1–4, 2013.

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

[33] J. Thiemann, N. Ito, and E. Vincent. The diverse environments multi-channel acoustic noise database (demand): A database of multichannel environmental noise recordings. *Journal of the Acoustical Society of America*, 133(5):3591–3591, 2013.

[34] 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.

[35] 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.

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

[37] C. Valentini-Botinhao, X. Wang, S. Takaki, and J. Yamagishi. Investigating RNN-based speech enhancement methods for noise-robust Text-to-Speech. In *Proc. 9th ISCA Workshop on Speech Synthesis Workshop (SSW 9)*, pages 146–152, 2016.

[38] E. Vincent, C. Fevotte, and R. Gribonval. Performance measurement in blind audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(4):1462–1469, 2006.

[39] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey. SDR – half-baked or well done? In *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2019.

[40] M. Kolbæk, Z.-H. Tan, S. H. Jensen, and J. Jensen. On Loss Functions for Supervised Monaural Time-Domain Speech Enhancement. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:825–838, 2020.

[41] 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.

[42] 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)*, volume 2, pages 749–752, 2001.

[43] H. Zhang, X. Zhang, and G. Gao. Training supervised speech separation system to improve STOI and PESQ directly. In *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 5374–5378, April 2018.

[44] D. L. Wang and J. Chen. Supervised Speech Separation Based on Deep Learning: An Overview. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(10):1702–1726, 2018.

[45] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis. Deep learning for monaural speech separation. In *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2014.

[46] 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)*, August 2015.

[47] A. Jansson, E. Humphrey, N. Montecchio, R. Bittner, A. Kumar, and T. Weyde. Singing voice separation with deep u-net convolutional networks. In *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, October 2017.

[48] 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.

[49] 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*, 27(8):1256–1266, 2019.

[50] T. von Neumann, K. Kinoshita, L. Drude, C. Boeddeker, M. Delcroix, T. Nakatani, and R. Haeb-Umbach. End-to-End Training of Time Domain Audio Separation and Recognition. In *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 7004–7008, 2020.

[51] Y. Luo, Z. Chen, and T. Yoshioka. Dual-path RNN: efficient long sequence modeling for time-domain single-channel speech separation. In *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2020.

[52] E. Tzinis, Z. Wang, and P. Smaragdis. Sudo rm -rf: Efficient Networks for Universal Audio Source Separation. In *Proc. of the IEEE Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, 2020.

[53] C. Subakan, M. Ravanelli, S. Cornell, F. Grondin, and M. Bronzi. Exploring self-attention mechanisms for speech separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:2169–2180, 2023.

[54] V. Zadorozhnyy, Q. Ye, and K. Koishida. SCP-GAN: Self-Correcting Discriminator Optimization for Training Consistency Preserving Metric GAN on Speech Enhancement Tasks. In *Proc. Interspeech*, pages 2463–2467, 2023.

[55] L. Liu, H. Guan, J. Ma, W. Dai, G. Wang, and S. Ding. A Mask Free Neural Network for Monaural Speech Enhancement. In *Proc. Interspeech*, pages 2468–2472, 2023.

[56] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[57] E. Strubell, A. Ganesh, and A. McCallum. Energy and Policy Considerations for Deep Learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, 2019.

[58] 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.

[59] 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.

[60] 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.

[61] 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.

[62] 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.

[63] 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)*, pages 16–20. IEEE, 2021.

[64] M. Delcroix, K. Žmolíková, K. Kinoshita, A. Ogawa, and T. Nakatani. Single Channel Target Speaker Extraction and Recognition with Speaker Beam. In *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 5554–5558, 2018.

[65] Q. Wang, H. Muckenhirn, K. Wilson, P. Sridhar, Z. Wu, J. R. Hershey, R. A. Saurous, R. J. Weiss, Y. Jia, and I. L. Moreno. VoiceFilter: Targeted Voice Separation by Speaker-Conditioned Spectrogram Masking. In *Proc. Interspeech*, pages 2728–2732, 2019.

[66] S. E. Eskimez, T. Yoshioka, H. Wang, X. Wang, Z. Chen, and X. Huang. Personalized Speech Enhancement: New Models and Comprehensive Evaluation. In *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 356–360, 2022.

[67] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli. Wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460, 2020.

[68] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed. HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460, 2021.

[69] A. Babu, C. Wang, A. Tjandra, K. Lakhotia, Q. Xu, N. Goyal, K. Singh, P. von Platen, Y. Saraf, J. Pino, A. Baevski, A. Conneau, and M. Auli. XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale. In *Proc. Interspeech*, pages 2278–2282, 2022.

[70] S. Watanabe, T. Hori, J. Le 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)*, pages 5275–5279. IEEE, 2017.

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

[72] 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)*, pages 5774–5778, 2021.

[73] 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)*, pages 436–440, 2021.

[74] 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.

[75] A. Sivaraman, S. Wisdom, H. Erdogan, and J. R. Hershey. Adapting Speech Separation to Real-World Meetings Using Mixture Invariant Training. In *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2022.

[76] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, March 1991.

[77] S.E. Chazan, J. Goldberger, and S. Gannot. Speech enhancement using a deep mixture of experts. *arXiv preprint arXiv:1703.09302*, 2017.

[78] 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.

[79] 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.

[80] P. Smaragdis. Non-negative matrix factor deconvolution: Extraction of multiple sound sources from monophonic inputs. In *Proceedings of International Conference on Independent Component Analysis and Blind Signal Separation*, pages 494–499, Granada, Spain, 2004.

[81] P. Smaragdis, B. Raj, and M. Shashanka. Supervised and semi-supervised separation of sounds from single-channel mixtures. In *Proc. of the International Conference on Independent Component Analysis and Blind Signal Separation (ICA)*, London, UK, 2007.

[82] T. O. Virtanen. Monaural sound source separation by perceptually weighted non-negative matrix factorization. Technical report, Tampere University of Technology, 2007.

[83] D. L. Sun and G. J. Mysore. Universal speech models for speaker independent single channel source separation. In *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2013.

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

[85] 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.

[86] 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*, 7(2):21S–25, 1998.

[87] K. Smeds, F. Wolters, and M. Rung. Estimation of Signal-to-Noise Ratios in Realistic Sound Scenarios. *Journal of the American Academy of Audiology*, 26(02):183–196, 2015.

[88] 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.

[89] 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)*, pages 369–374, 2013.

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

[91] 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)*, pages 334–340, 2018.

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

[93] 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)*, pages 8024–8035. 2019.

[94] M. Pariente et al. Asteroid: The PyTorch-Based Audio Source Separation Toolkit for Researchers. In *Proc. Interspeech*, 2020.

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

[96] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[97] S.E. Chazan, J. Goldberger, and S. Gannot. Deep recurrent mixture of experts for speech enhancement. In *Proc. of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 359–363. IEEE, 2017.

[98] E. Agustsson, F. Mentzer, M. Tschannen, L. Cavigelli, R. Timofte, L. Benini, and L. V. Gool. Soft-to-hard vector quantization for end-to-end learning compressible representations. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1141–1151, 2017.

[99] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet. Front-End Factor Analysis for Speaker Verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798, 2010.

[100] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur. Deep Neural Network Embeddings for Text-Independent Speaker Verification. In *Proc. Interspeech*, pages 999–1003, 2017.

[101] K. Žmolíková, M. Delcroix, K. Kinoshita, T. Higuchi, A. Ogawa, and T. Nakatani. Learning speaker representation for neural network based multichannel speaker extraction. In *Proc. of the IEEE Automatic Speech Recognition and Understanding Workshop*, pages 8–15, 2017.

[102] K. Chen and A. Salman. Learning speaker-specific characteristics with a deep neural architecture. *IEEE Transactions on Neural Networks*, 22(11):1744–1756, 2011.

[103] A. Jati and P. G. Georgiou. Speaker2Vec: Unsupervised Learning and Adaptation of a Speaker Manifold Using Deep Neural Networks with an Evaluation on Speaker Segmentation. In *Proc. Interspeech*, pages 3567–3571, 2017.

[104] Y. Koizumi, K. Yatabe, M. Delcroix, Y. Masuyama, and D. Takeuchi. Speech Enhancement Using Self-Adaptation and Multi-Head Self-Attention. In *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2020.

[105] F.-K. Chuang, S.-S. Wang, J.-W. Hung, Y. Tsao, and S.-H. Fang. Speaker-Aware Deep Denoising Autoencoder with Embedded Speaker Identity for Speech Enhancement. In *Proc. Interspeech*, 2019.

[106] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah. Signature verification using a “siamese” time delay neural network. In *Advances in Neural Information Processing Systems (NIPS)*, pages 737–744, 1994.

[107] A. Narayanan and D. L. Wang. Ideal ratio mask estimation using deep neural networks for robust speech recognition. In *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2013.

[108] L. Van der Maaten and G. Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(11), 2008.

[109] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[110] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[111] G. Wichern, J. Antognini, M. Flynn, L. R. Zhu, E. McQuinn, D. Crow, E. Manilow, and J. Le Roux. WHAM!: Extending Speech Separation to Noisy Environments. In *Proc. Interspeech*, pages 1368–1372, 2019.

CURRICULUM VITAE

EDUCATION

Ph.D., Intelligent Systems Engineering, **Indiana University**	*May 2024*
B.S., Electrical Engineering, **University of Illinois at Urbana-Champaign**	*May 2015*

WORK EXPERIENCE

AI/ML Resident, **Apple**	*Jul 2023 – current*
Research Intern, **Microsoft**	*Jul 2022 – Sep 2022*
Research Intern, **Google**	*May 2021 – Oct 2021*
AI Resident, **X Development LLC.**	*May 2020 – Dec 2020*
Research Scientist Intern, **Spotify**	*Jun 2019 – Aug 2019*
Applied Scientist Intern, **Amazon**	*May 2018 – Aug 2018*
Embedded Software Engineer, **Qualcomm**	*Jul 2015 – Mar 2017*

TEACHING EXPERIENCE

ENGR-E511: “Machine Learning For Signal Processing”	*2023 Spring*
ENGR-E511: “Machine Learning For Signal Processing”	*2021 Spring*
ENGR-E533: “Deep Learning Systems”	*2020 Fall*
ENGR-E511: “Machine Learning For Signal Processing”	*2020 Spring*
ENGR-E533: “Deep Learning Systems”	*2019 Fall*
CS 498 RK: “The Art and Science of Web Programming”	*2015 Spring*

SKILLS

Programming: Python, C, C++, MATLAB

Frameworks: PyTorch, Tensorflow, Keras

Web Development: HTML, CSS, PHP, JavaScript, SQL, NoSQL, MongoDB

CONFERENCE / JOURNAL REVIEWER

- IEEE Journal of Selected Topics in Signal Processing (JSTSP)
- IEEE Transactions on Audio, Speech and Language Processing (TASLP)
- Speech Communication
- IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)
- International Society for Music Information Retrieval Conference (ISMIR)
- European Signal Processing Conference (EUSIPCO)
- ACM SIGCHI Conference on Designing Interactive Systems (DIS)
- AAAI National Conference on Artificial Intelligence (AAAI)

SELECT PUBLICATIONS

International Journal Articles

[J-1] A. Sivaraman and M. Kim, “Efficient Personalized Speech Enhancement Through Self-Supervised Learning,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1342–1356, Oct. 2022.

Refereed International Conference Proceedings

[C-1] A. Sivaraman and M. Kim, “Sparse Mixture of Local Experts for Efficient Speech Enhancement,” in *Proc. Interspeech*, Sep. 2020, pp. 4526–4530.

[C-2] S. Reddy, Y. Yu, A. Pappu, A. Sivaraman, R. Rezapour, and R. Jones, “Detecting Extraneous Content in Podcasts,” in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, Apr. 2021, pp. 1166–1173.

[C-3] A. Sivaraman, S. Kim, and M. Kim, “Personalized Speech Enhancement through Self-Supervised Data Augmentation and Purification,” in *Proc. Interspeech*, Sep. 2021, pp. 2676–2680.

[C-4] 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)*, Oct. 2021, pp. 171–175.

[C-5] A. Sivaraman, S. Wisdom, H. Erdogan, and J. R. Hershey, “Adapting Speech Separation to Real-World Meetings Using Mixture Invariant Training,” in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2022, pp. 686–690.

[C-6] A. Kuznetsova, A. Sivaraman, and M. Kim, “The Potential of Neural Speech Synthesis-Based Data Augmentation for Personalized Speech Enhancement,” in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2023.

Workshops

[W-1] A. Sivaraman and M. Kim, “Self-Supervised Learning from Contrastive Mixtures for Personalized Speech Enhancement,” in *Advances in Neural Information Processing Systems (NeurIPS), Workshop on Self-Supervised Learning for Speech and Audio Processing*, Dec. 2020

PATENTS

`US11416742B2` Audio signal encoding method and apparatus and audio signal decoding method and apparatus using psychoacoustic-based weighted error function

`US11234031B1` Systems and methods for skip-based content detection

ProQuest Number: 31243527

INFORMATION TO ALL USERS
The quality and completeness of this reproduction is dependent on the quality
and completeness of the copy made available to ProQuest.

Distributed by ProQuest LLC (2024).
Copyright of the Dissertation is held by the Author unless otherwise noted.

This work may be used in accordance with the terms of the Creative Commons license
or other rights statement, as indicated in the copyright statement or in the metadata
associated with this work. Unless otherwise specified in the copyright statement
or the metadata, all rights are reserved by the copyright holder.

This work is protected against unauthorized copying under Title 17,
United States Code and other applicable copyright laws.

Microform Edition where available © ProQuest LLC. No reproduction or digitization
of the Microform Edition is authorized without permission of ProQuest LLC.

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346 USA