ZERO-SHOT PERSONALIZED SPEECH ENHANCEMENT THROUGH SPEAKER-INFORMED MODEL SELECTION

*Aswin Sivaraman, Minje Kim**

Indiana University, Department of Intelligent Systems Engineering, USA

asivara@indiana.edu, minje@indiana.edu

ABSTRACT

This paper presents a novel zero-shot learning approach towards personalized speech enhancement through the use of a sparsely active ensemble model. Optimizing speech denoising systems towards a particular test-time speaker can improve performance and reduce run-time complexity. However, test-time model adaptation may be challenging if collecting data from the test-time speaker is not possible. To this end, we propose using an ensemble model wherein each specialist module denoises noisy utterances from a distinct partition of training set speakers. The gating module inexpensively estimates test-time speaker characteristics in the form of an embedding vector and selects the most appropriate specialist module for denoising the test signal. Grouping the training set speakers into non-overlapping semantically similar groups is non-trivial and ill-defined. To do this, we first train a Siamese network using noisy speech pairs to maximize or minimize the similarity of its output vectors depending on whether the utterances derive from the same speaker or not. Next, we perform k-means clustering on the latent space formed by the averaged embedding vectors per training set speaker. In this way, we designate speaker groups and train specialist modules optimized around partitions of the complete training set. Our experiments show that ensemble models made up of low-capacity specialists can outperform high-capacity generalist models with greater efficiency and improved adaptation towards unseen test-time speakers.

Index Terms— Speech enhancement, deep learning, adaptive mixture of local experts, model compression by selection

1. INTRODUCTION

Speech enhancement (SE) is a long-standing research area within signal processing [1] which has experienced significant progress in the past decade due to the pervasiveness of machine learning models and deep neural networks (DNNs) [2, 3, 4]. This paper addresses the task of removing non-stationary background noise from a monophonic recording of a single speaker. The majority of research in this area proposes solutions to generalized speech enhancement—in other words, the denoising models make no assumptions about the test-time speaker or noisy environment. Many DNN-based general-purpose SE models require millions of learnable parameters to address the diversity in speakers and noise types. However, given the ubiquity of resource-constrained devices (i.e., smartphones or smart speakers), we focus our interest on developing SE models which minimize run-time computational complexity without sacrificing denoising quality.

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

As it was previously shown that models addressing a disjoint sub-problem outperform models trained towards universal speech enhancement [5], classifying a noisy test signal as belonging to a specific sub-problem may improve the performance of a modular SE system. One type of divide-and-conquer algorithm design paradigm was introduced to neural networks, referred to as the "mixture of local experts" (MLoE) architecture [6]. MLoE, or ensemble models, consist of independent expert modules (i.e., specialist networks), each trained to address a subset of all the training cases. Consequently, an auxiliary classifier module (i.e., gating network) is trained to estimate a weighting over all the local experts based on their relevance towards a particular input.

Recent research has explored different ways of applying the MLoE paradigm towards speech enhancement. One paper poses local experts associated with varying choices of hyperparameter (e.g., with different contextual window lengths) [7]. Another study composes an MLoE model using recurrent cells, which can model the temporal structure inherent to speech [8]. In these works, the local experts are not trained to address specific sub-problems. However, a more recent MLoE-based SE system showed substantial improvements using two predefined partitioning schemes: based on the quality of input signal, i.e., in terms of signal-to-noise ratio (SNR), and the gender of the speakers [9]. Furthermore, by introducing "sparseness" to the gating network's weights, this recent paper performs test-time inference using only the most suitable specialist. Compared to generalist models, which require a large model capacity to achieve a certain level of speech denoising, an ensemble model can yield the same enhancement quality even if the composing specialists use much fewer parameters. Therefore, we claim that a sparse ensemble of specialists is a form of model compression [9]. One article proposed a similar partitioning strategy based on the speech quality, but by employing a speech quality estimator in place of the traditional gating module [10]. This work also suggested using learned SNR-based partitions as opposed to predefined partitions. Most recently, Chazan et al. introduced an MLoE model for SE by defining sub-problem based on clustering of clean speech [11].

In this paper, we investigate using MLoE as a means for *personalizing* an SE model. To achieve this, we propose learning the optimal speaker grouping from the noisy utterances, in contrast to the previously mentioned works. Once speaker groups are defined, 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. This idea of "zero-shot" speech enhancement through model selection has seen some preliminary assessment. However, the prior research applied limited model selection based on speaker-agnostic aspects of the

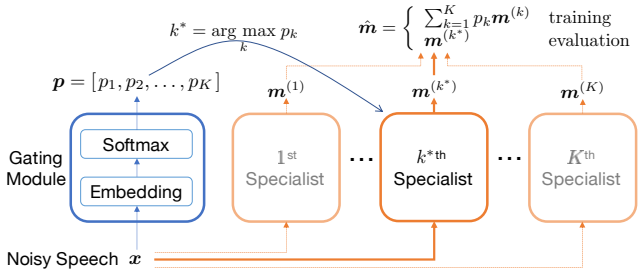


Figure 1: The proposed sparse ensemble of specialists model.

signals, such as quality of the test-time signal, e.g., its signal-to-noise (SNR) level, [10, 9], types of the noise sources [12], or speaker gender [9].

Another important aspect of our work confronts an 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* [13] or *x-vectors* computed using a time-delay neural network [14]. Prior works have also used sequence summarizing networks [15] either through contrastive loss [16] or by estimating subsequent frames for a single input signal [17]. Although these learn valid speaker-identifying features, we propose a custom embedding-learning model which can effectively function as the gating network as in [11]. Additionally, we want our custom embedding to be robust to additive noise; previously proposed noise-robust embedding vectors [18, 19, 20] were not designed around MoLE. To do this, we develop a Siamese network [21], 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 [9], our zero-shot sparse ensemble model for personalized SE minimizes test-time computational complexity.

2. PROPOSED METHODS

2.1. Ensemble Models

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 [9, 10], 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.

We illustrate the architecture of a sparsely active ensemble model in Figure 1. 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) [22] 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.

2.2. 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).

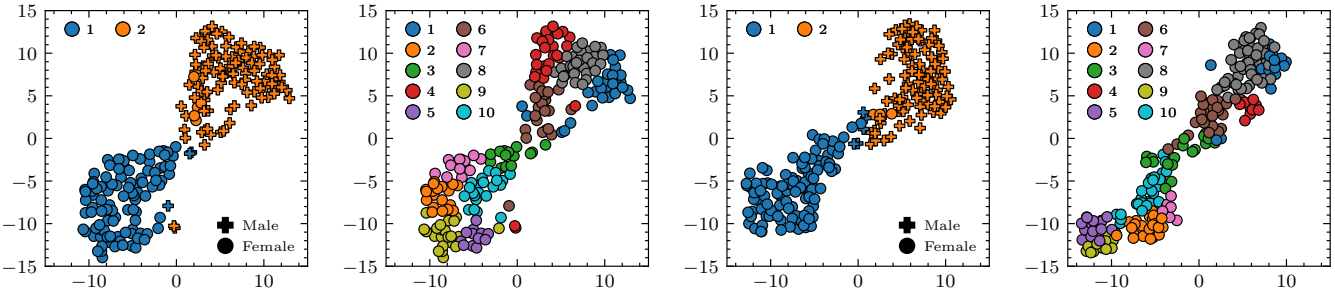
We can then generate pairs of noisy speech utterances, $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. Once f predicts the embeddings, i.e., $\boldsymbol{z}_i = f(\boldsymbol{x}_i)$ and $\boldsymbol{z}_j = f(\boldsymbol{x}_j)$, their inner product serves as a measure of similarity. A sigmoid function follows 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 [21] 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.

2.3. 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.

Figure 2 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 $= 40$) [23]. These subplots show that the SV model



(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 2: Subplots comparing various choices of K for using k-means clustering on the speaker embeddings. 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 (a) and 10 in (b). After fine-tuning an ensemble model, the gating network's embedding function f adjusts its parameters towards the speech enhancement (SE) objective. The latent space is modified uniquely based on the ensemble's configuration. In (a) and (b), the class labels derive from k-means clustering, but in (c) and (d) the class labels are estimated by the gating network's classifier function g.

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.

2.4. 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 Sec. 2.6 and propose a fine-tuning solution.

2.5. 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 Sec. 2.3. 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 (SI-SDR) [24] as the loss function.

2.6. 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 $k^* = \operatorname{argmax}_k p_k$. 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 [9].

Figure 2c and 2d 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.

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 [25] *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 [26], 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

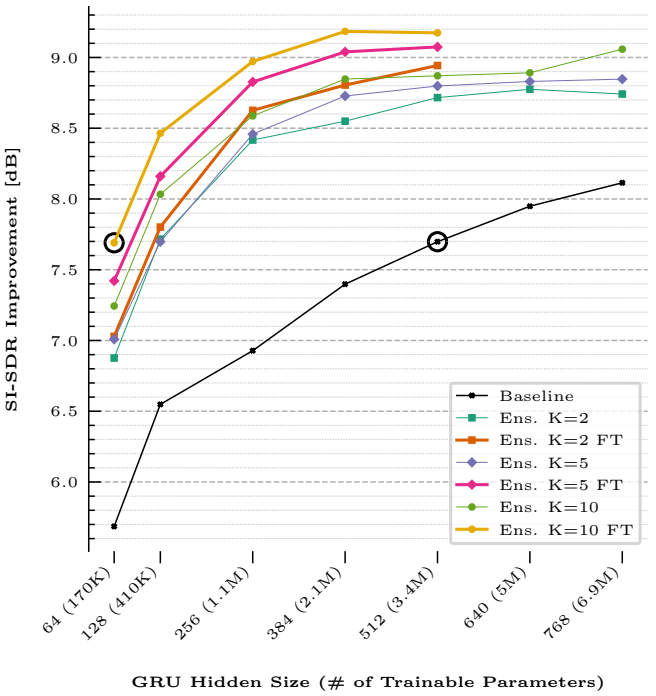


Figure 3: Comparison of speech enhancement performance between a baseline general-purpose model against different configurations of speaker-informed sparse ensemble models.

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 [27]. 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. RESULTS

Figure 3 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 Figure 3, 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 SI-SDR improvement achieved with the ensemble models increases along with K from 2 to 5 to 10 through fine-tuning.

5. CONCLUSION

We investigated model adaptation through selection (the "mixture of local experts" paradigm) as a means for personalized speech enhancement. Our method is zero-shot as the system 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. Source code and sound examples can be found at: `https://saige.sice.indiana.edu/research-projects/sparse-mle/`

6. ACKNOWLEDGEMENT

The authors appreciate the discussion with Francesco Nesta during the initial phase of the work.

7. REFERENCES

[1] S. F. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 27, pp. 113–120, 1979.

[2] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "An experimental study on speech enhancement based on deep neural networks," *IEEE Signal Processing Letters*, vol. 21, no. 1, pp. 65–68, 2014.

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

[4] D. L. Wang and J. Chen, "Supervised speech separation based on deep learning: An overview," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 10, pp. 1702–1726, 2018.

[5] M. Kolbæk, Z. H. Tan, and J. Jensen, "Speech intelligibility potential of general and specialized deep neural network based speech enhancement systems," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 1, pp. 153–167, Jan 2017.

[6] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural Computation*, vol. 3, no. 1, pp. 79–87, Mar. 1991.

[7] X.-L. Zhang and D. Wang, "A deep ensemble learning method for monaural speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 5, pp. 967–977, 2016.

[8] S. Chazan, J. Goldberger, and S. Gannot, "Deep recurrent mixture of experts for speech enhancement," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2017, pp. 359–363.

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

[10] 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.

[11] S. E. Chazan, J. Goldberger, and S. Gannot, "Speech Enhancement with Mixture of Deep Experts with Clean Clustering Pre-Training," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2021, pp. 716–720.

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

[13] 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*, vol. 19, no. 4, pp. 788–798, 2010.

[14] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep Neural Network Embeddings for Text-Independent Speaker Verification," in *Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech)*, 2017, pp. 999–1003.

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

[16] K. Chen and A. Salman, "Learning speaker-specific characteristics with a deep neural architecture," *IEEE Transactions on Neural Networks*, vol. 22, no. 11, pp. 1744–1756, 2011.

[17] 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 *Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech)*, 2017, pp. 3567–3571.

[18] Y. Koizumi, K. Yatabe, M. Delcroix, Y. Masuyama, and D. Takeuchi, "Speech Enhancement Using Self-Adaptation and Multi-Head Self-Attention," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2020, pp. 181–185.

[19] 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 *Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech)*, 2019, pp. 3173–3177.

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

[21] 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)*, 1994, pp. 737–744.

[22] A. Narayanan and D. L. Wang, "Ideal ratio mask estimation using deep neural networks for robust speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2013, pp. 7092–7096.

[23] L. V. der Maaten and G. Hinton, "Visualizing Data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. 11, 2008.

[24] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "SDR – half-baked or well done?" in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 626–630.

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

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

[27] 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.