Interspeech 2024
1-5 September 2024, Kos, Greece

Personalized Speech Enhancement Without a Separate Speaker Embedding Model

Tanel Pärnamaa, Ando Saabas

Microsoft Corporation
tanel.parnamaa@microsoft.com

Abstract

Personalized speech enhancement (PSE) models can improve the audio quality of teleconferencing systems by adapting to the characteristics of a speaker's voice. However, most existing methods require a separate speaker embedding model to extract a vector representation of the speaker from enrollment audio, which adds complexity to the training and deployment process. We propose to use the internal representation of the PSE model itself as the speaker embedding, thereby avoiding the need for a separate model. We show that our approach performs equally well or better than the standard method of using a pre-trained speaker embedding model on noise suppression and echo cancellation tasks. Moreover, our approach surpasses the ICASSP 2023 Deep Noise Suppression Challenge winner by 0.15 in Mean Opinion Score.

Index Terms: speech enhancement, personalized speech enhancement, target speech extraction, real-time processing

1. Introduction

Audio signal enhancement components, such as noise suppression (NS), dereverberation and acoustic echo cancellation (AEC), are essential elements of modern teleconferencing systems. While they significantly improve speech signal quality, they have some limitations. For example, they are unable to remove unwanted human voices from the background, which are common in open office spaces or cafeteria environments. Additionally, echo cancellers typically fail to remove echoes resulting from delays that are too long or non-causal between the far end and microphone signals. By knowing the user's voice, it is possible to enhance audio quality even in these challenging scenarios by isolating the user's voice from the input signal.

The standard approach to personalized speech enhancement (PSE) is based on the work on speaker-conditioned single-speaker extraction [1, 2, 3]. It consists of multiple stages. Firstly, a pre-trained speaker embedding model is used to extract an embedding vector from enrollment audio, representing the characteristics of a person's voice. Then, this embedding is used in a separate speech extraction or enhancement model. The speaker embedding model can be fixed, fine-tuned or trained jointly with the speech enhancement model.

Unfortunately, the multi-stage and multi-model approach has many practical difficulties: separate models need to be trained, deployed, maintained, and kept in synchronization, resulting in a significant engineering overhead, especially when deploying the models for edge devices.

In contrast, we propose an approach that does not rely on a separate embedding model. In particular, we note that for the speech enhancement model to make use of the speaker embedding, it needs to internally compute the representation of the speaker's voice in the input data that it is applied to, and compare it to the given embedding. Instead of having a separate speaker embedding model, we make use of this internal embedding to characterize a speaker's voice profile, and therefore only need a single model that is responsible for both speech enhancement and extraction of the speaker embedding. This change simplifies the training and deploying of personalized models.

This approach also offers other benefits, such as simplifying the enrollment process. In a typical personalized model usage scenario, the user must first provide an enrollment audio clip, for example, by reading a piece of text and recording it. Capturing enrollment audio automatically is more appealing because it removes the initial friction of using personalized models. However, speaker embedding models are usually too large and slow to run continuously on client devices. Running the speaker embedding model on a server might raise privacy concerns. By extracting the embedding directly from the speech enhancement model already in use for audio quality enhancement, we simplify the auto-enrollment process and minimize computational requirements because only one model is necessary.

We start with the state-of-the-art (SOTA) speech enhancement model, DeepVQE [4]. Firstly, we personalize it using the standard approach with a large pre-trained speaker embedding model. Next, we train the personalized model from scratch using its internal representation for speaker embedding, without changing its architecture or complexity. Our results show that this method matches the performance of the two-stage approach. Both models achieve SOTA results on the DNS Challenge noise suppression test data.

2. Related work

Recently, several challenges have been organized to benchmark methods for personalized NS and AEC. The top entries for the ICASSP 2022 and 2023 Deep Noise Suppression (DNS) Challenge personalized tracks utilized a two-stage approach with a separate embedding model for enrollment [5, 6, 7, 8, 9, 10]. Similarly, the winning entry for the ICASSP 2023 AEC Challenge used a separate embedding model [11, 12].

A common design choice in the two-stage approach is whether to use a fixed speaker embedding model, as done in VoiceFilter [1] and PercepNet [13], or jointly learn or finetune the embedder, as done in SpeakerBeam [3] and SpEx [14]. Liu et al. showed in [15] that fine-tuning can improve audio quality on in-domain data, but the improvement does not generalize to out-of-domain data. Moreover, none of the top participants in the DNS challenge finetuned the speaker embedding model, supporting the observation that transfer from simulated training data to real world test data is challenging with finetuned models. This is unexpected because the representation learned for

10.21437/Interspeech.2024-1234

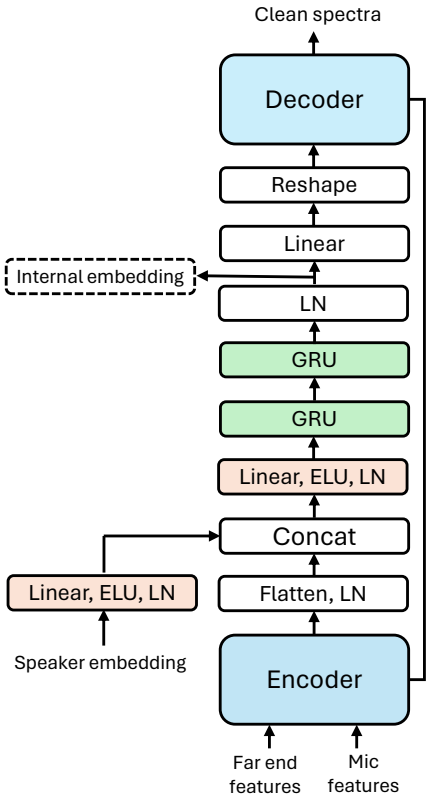

Figure 1: *Model architecture with speaker information fusion. The figure shows how the speaker embedding is concatenated with the encoder features, the details of the temporal block, and the location of the internal embedding that we use to characterise speakers.*

the speaker verification task, which is commonly used to train the embedding model, may not be optimal for the speech enhancement task.

In [15], the authors questioned the need for large speaker embedding models. They analysed the role of speaker embeddings in target speaker separation and found that the log-mel filterbank features worked surprisingly well on cross-dataset evaluation compared to learned features. However, they only evaluated on complete mixtures, and overlooked aspects like oversuppression and background speaker leakage, which are critical in teleconferencing systems.

Different to the standard approach, Sivaraman et al. used a method involving a mixture of local experts that does not require a reference speech utterance [16]. During inference, a separate gating module embeds the audio and selects a specialised expert module that best fits the speaker. However, the training process involves a complex clustering based on speaker embeddings and multiple pre-training stages.

3. Proposed method

As a starting point, we take DeepVQE, a SOTA speech enhancement model for joint NS, AEC and dereverberation [4]. This model shows best-in-class performance in the ICASSP 2023 AEC Challenge [11] and the non-personalized track of the ICASSP 2023 DNS Challenge [6]. The main building blocks of the model are an encoder, a Gated Recurrent Unit (GRU) bottleneck, a decoder, and a deep filter for output reconstruction called a complex convolving mask block. The input features are power law compressed complex spectra. The far end and microphone features are encoded separately, and then soft-aligned using a cross-attention mechanism. The decoder uses sub-pixel convolutions for upsampling. For more details, we refer the reader to [4]. We will condition the model based on speaker embedding to make it personalized, and show how to use its internal embedding to describe enrollment audio.

3.1. Personalized DeepVQE

To make the model personalized and suppress neighboring talkers, we need to fuse speaker information to the model. We first describe the standard two-stage approach to personalization using a separate pre-trained embedding model. In Section 3.2, we show how we can remove the separate embedding model, without requiring any changes to the speech enhancement models.

In the two-stage approach, a speaker embedding vector is extracted from enrollment audio using a pre-trained model, then linearly projected to match the size of flattened features before the GRU layer, followed by an activation function and layer normalization (LN) [17]. The output is then concatenated with the flattened features, and linearly transformed to match the original size of the flattened features, which is again followed by an activation function and LN. The fusion is shown in Figure 1.

We use a different residual block and temporal block compared to the DeepVQE architecture. Specifically, we adopt an inverted residual block [18], but drop the squeeze and excitation block [19], and use standard convolutions instead of depthwise convolutions. Moreover, we use two GRU layers, and layer-normalize the input of the first GRU layer and the output of the last GRU layer. The temporal block is shown in Figure 1. These changes showed improvement over the baseline DeepVQE architecture in the PSE scenario.

3.2. Proposed embedding extraction

To achieve personalization, the PSE model has to represent speaker information in its internal states. Intuitively, when the input signal contains a single voice, the PSE model compares its representation of the current speaker with the enrollment embedding to decide whether to suppress the speech. How can this speaker information be extracted from the PSE model? Most speech enhancement models employ an encoder-decoder approach with a temporal block in between. A natural place to extract an internal embedding is from the temporal component. For example, models like DeepVQE, PercepNet, VoiceFilter-Lite, and E3Net [20] employ an RNN cell, from which the internal embedding can be extracted. In transformer-based models such as MTFAA-Net [21], the internal embedding can be obtained after a specified transformer layer.

In our experiments, we extract the internal embedding after the temporal block, and average the frame-based embeddings gathered from enrollment audio. Specifically, we extract T frames of features from enrollment audio, which produce T embeddings of size K. We then average these to get a single embedding of size K. This embedding may also contain other information, such as the presence of echo or noise categories, but the PSE model will learn to use information from the state that is conducive for minimizing the loss.

The internal embedding could also be gathered from other locations in the PSE model, such as before the temporal block or between the GRU layers. Moreover, when taking the internal embedding after the fusion of speaker information, we also need to specify the initial speaker embedding when mapping

enrollment audio to a speaker embedding. We use a vector of zeros for that.

We want to emphasize that this change, moving the embedding extraction from a separate model to the speech enhancement model itself, does not require any change in the speech enhancement model. Both the architecture and the size of the enhancement model remain the same in both approaches.

4. Experimental setup

To understand the effectiveness of the proposed method, we compare it to multiple baseline models. Firstly, we use a model without a speaker embedding, representing a scenario where the closest speaker to the microphone is extracted from the audio signal. This condition helps to assess the overall usefulness of speaker information. It is implemented by setting the speaker embedding to zeros in the personalized DeepVQE model. Secondly, we use log-mel filterbank features as the speaker embedding to understand the improvement of learned features over simple feature extraction. For this, we follow [15] and compute 80-dim FBANK features, and concatenate the temporal mean and standard deviation to get an embedding of size 160. Thirdly, we use a Res2Net model trained for speaker verification to extract speaker features [22].

4.1. Training data

We adopt the data generation approach outlined in [4], with modifications to include enrollment clips and utilize background speech in addition to background noises. Beyond the datasets provided in the ICASSP 2023 AEC and DNS challenges, we also use VoxCeleb2 [23], which we pre-process with a noise suppressor to remove background noises. As the speaker identity information can be wrong or clips might contain multiple speakers, we remove clips where intra-speaker distance of clip embeddings is high [24]. The sampled enrollment clips are 10 seconds long, and 50% of them are noisy with an SNR sampled from the range [0, 40]. The training clips are 40 seconds in length, and 30% of the clips contain background speech with an SIR sampled from the range [0, 20].

4.2. Evaluation data

For evaluation, we use the AEC Challenge 2023 blind test set to assess AEC performance, DNS Challenge 2023 blind test set to evaluate personalized NS performance, AMI dataset [25] for evaluating target speaker over-suppression. The AEC and DNS test set are used as provided in the challenges. From the AMI dataset, we sampled 1549 clips where only a single speaker is present, and ensured enrollment audio was recorded on a different device. This is important because most over-suppression cases happen when the target speech differs from the enrollment, such as when the audio is recorded from a different location or using a different microphone, which are common scenarios in practice. The average test clip length is 23.8 seconds, and the average enrollment length is 12.2 seconds. For the evaluation of background speech suppression, we sampled clips from LibriVox data such that the enrollment speaker and nearend speaker are different. The perfect outcome is to suppress all content in such clips. Additionally, we use internally collected clips where the target speaker is present at the beginning, followed by a long period of background talk. We use 220 clips in total for background speech suppression evaluation, with an average clip length of 42.3 seconds and an average enrollment length of 12.6 seconds. Furthermore, we created 200 synthetic 15-second mixtures based on LibriVox data to evaluate target speaker extraction with reference-based metrics.

4.3. Hyperparameters

We use a 20ms squared root Hann window, a hop size of 10ms, a discrete Fourier transform length of 320, and sample audio at 16 kHz. We extract power law compressed complex spectra from the noisy microphone input and far end signal. If the far end signal is missing, we set the signal to zero.

We adapt the DeepVQE-S configuration for comparison study, and name the personalized configuration as PVQE-S. We use two encoder blocks for the far end and microphone signal, followed by an alignment block with a history of 100 frames to align the features in time with a maximum delay of 1 second. The far end branch uses 8 and 24 filters, and the microphone branch uses 16 and 40 filters. The microphone features and aligned far end features are concatenated, and fed into a combined encoder that consists of two blocks of 56 and 24 filters. The kernel size for the encoder is 2x3, with the first dimension representing temporal and the second dimension frequency axis. For the temporal block, we use two GRU layers of size 256 followed by a linear projection. Moreover, we layer-normalize the input and output of the temporal block. The decoder uses subpixel convolutions and has 4 blocks of 40, 32, 32, and 27 filters. The combined encoder and the first two decoder blocks make use of a residual block with an expansion factor of 0.7 [18]. Exponential linear unit (ELU) is used as an activation function throughout the model [26]. No look-ahead is used in the model.

The speaker embedding is fused to the model before the temporal block. Specifically, a linear layer transforms the speaker embedding to a size of 240, followed by an activation function and LN. The output is then concatenated with the flattened features from the encoder and projected back to the size of the flattened features. In experiments with a pre-trained speaker embedding model, we use a Res2Net-based speaker verification model [22], which returns an embedding of size 128 and has 15M parameters. In the proposed method, we extract the internal embedding from the output of LN which comes after the last GRU layer.

The models are trained using the Adam optimizer with a learning rate of 6×10^{-5} and a weight decay of 1×10^{-7} for 234k iterations on batches of 64 samples. A complex compressed MSE loss [27] is used with an exponent of 0.3 and a beta of 0.7.

5. Results

5.1. Objective evaluation

For objective evaluation, we use non-intrusive neural network-based mean opinion score (MOS) estimators, personalized DNSMOS P.835 [28] and AECMOS [29], to assess the quality of noise and echo removal. Personalized DNSMOS P.835 provides three scores: speech quality (SIG), background noise quality (BAK), and overall quality (OVRL) of the audio. AECMOS gives two scores: echo removal (AECMOS Echo) and signal degradation (AECMOS Deg) quality. Additionally, we report echo return loss enhancement (ERLE) for the far end single talk scenario and Perceptual Evaluation of Speech Quality (PESQ) [30] for simulated two-speaker mixtures. Furthermore, we report the target speaker over-suppression metric (TSOS) from [20]. Finally, to evaluate the effectiveness of the models in scenarios where only interfering speakers are present, we measure the signal energy reduction in decibels, where a higher

Table 1: *Objective quality comparison with baselines for AEC in far end single talk and double talk scenarios, personalized NS, target speaker over-suppression, and background speakers removal quality. All models use the PVQE-S architecture except the Noisy baseline.*

Speaker embedding	AECMOS Echo ↑	ERLE ↑	AECMOS Echo ↑	AECMOS Deg ↑	SIG ↑	BAK ↑	OVRL ↑	PESQ ↑	TSOS ↓	BAK SUPPR ↑
(Noisy)	1.99	0.0	1.79	3.85	4.16	2.27	2.65	1.61	**0.000**	0.00
Empty	4.58	68.9	4.55	**3.98**	4.03	3.97	3.53	2.68	0.003	5.37
FBANK	4.63	74.7	4.56	3.96	4.03	3.97	3.53	2.66	0.015	16.96
Separate	4.64	74.9	4.57	3.97	4.02	4.04	3.56	**2.72**	0.027	**34.30**
Internal	**4.68**	**75.7**	**4.61**	3.92	4.01	**4.13**	**3.59**	2.67	0.010	33.04

Table 2: *Subjective MOS results on the DNS challenge blind test set. We include the challenge winner and personalized baseline. The topmost three models are larger, and the rest are smaller models.*

		Track 1 - Headset			Track 2 - Speakerphone				
Model	Embedding	SIG	BAK	OVRL	SIG	BAK	OVRL	Avg	CI
PVQE-L	Internal	**3.61**	**3.10**	**2.96**	**3.70**	**3.12**	**3.00**	**3.25**	0.04
PVQE-L	Separate	3.58	3.03	2.89	**3.70**	3.09	2.95	3.21	0.04
Challenge winner [8]	Separate	3.58	2.87	2.75	3.69	2.90	2.83	3.10	0.04
PVQE-S	Internal	**3.47**	**2.70**	**2.58**	**3.59**	**2.76**	**2.68**	**2.96**	0.04
PVQE-S	Separate	**3.47**	2.64	2.52	3.48	2.60	2.54	2.88	0.04
Challenge baseline (E3Net [20])	Separate	3.28	2.62	2.46	3.50	**2.76**	2.64	2.87	0.04
Noisy	-	3.62	1.28	1.29	3.71	1.25	1.25	2.06	0.02

number indicates better performance (BAK SUPPR).

Table 1 shows the objective metrics for original noisy data, baseline methods and proposed models. Firstly, we can see that the use of speaker embedding gives a clear performance boost in background speech suppression scenarios. This is indicated by the BAK and BAK SUPPR scores, where models using the embedding outperform the no-embedding and filter-bank based approaches, without degrading signal scores.

At the same time, we see that the model using an internal embedding gives similar results to the two-stage model, even outperforming it for some metrics. Especially noteworthy is that while the internal embedding model gives as good or better results for background speech removal, it does not do so at the cost of near-end over-suppression. In fact, the opposite is true, the TSOS metric shows 2.7x reduction in over-suppressed frames.

5.2. Subjective evaluation results

To evaluate our approach with respect to the state of the art, we compare it against the baseline and winning models of the 2023 DNS Challenge. For a fair comparison, we increased the model size while maintaining the real-time constraints specified by the challenge rules. Specifically, we increased the layer sizes to match the base configuration in [4] and trained the large models on super-wideband data.

We conducted a subjective evaluation on the DNS challenge data using the personalized version of ITU-T P.835 framework [31], which uses 5 seconds of clean enrollment speech from primary talkers to help human raters recognize the primary speaker's voice when scoring the clips. The evaluation was run on a crowd-sourcing platform with 10 raters per clip.

Our large models (PVQE-L) outperformed the challenge winner, achieving a higher BAK score of over 0.2 MOS on both tracks without compromising signal quality. For large models, both the pre-trained and the internal speaker embedding extraction yielded similar results. However, for small models, the internal embedding method was more effective, improving the track 2 overall score by 0.14. This demonstrates that using an internal embedding is especially useful for small real-time models, leading to a simple and effective approach for balancing the representations given by a speaker embedding model and the PSE model.

5.3. Inference speed

We measure the inference speed of the large and small model on an Intel Core i7 10700K@3.8GHz CPU. We use a single-thread configuration and report the average inference time over 100,000 frames. The large model has 8.38M parameters and takes 3.64ms per frame, while the small model has 1.07M parameters and takes only 0.135ms per frame. This means that the small model can process audio signals 74 times faster than real-time, achieving a real-time factor of 0.0135. The small model demonstrates a remarkable balance between performance and complexity, making it suitable for real-time teleconferencing applications.

6. Conclusions

In this paper, we proposed a novel and simple approach to personalized speech enhancement that does not require a separate speaker embedding model. We showed that the internal representation of the speech enhancement model can be used as the speaker embedding. We evaluated our approach on two speech enhancement tasks: noise suppression and echo cancellation, and showed that the model achieves state-of-the-art performance on the DNS Challenge data. We compared our approach with the standard method of using a pre-trained speaker embedding model, and found that our approach improves background noise quality for small models.

7. References

[1] 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 2019*, 2019, pp. 2728–2732.

[2] Q. Wang, I. L. Moreno, M. Saglam, K. Wilson, A. Chiao, R. Liu, Y. He, W. Li, J. Pelecanos, M. Nika, and A. Gruenstein, "VoiceFilter-Lite: Streaming Targeted Voice Separation for On-Device Speech Recognition," in *Proc. Interspeech 2020*, 2020, pp. 2677–2681.

[3] K. Žmolíková, M. Delcroix, K. Kinoshita, T. Ochiai, T. Nakatani, L. Burget, and J. Černockỳ, "SpeakerBeam: Speaker aware neural network for target speaker extraction in speech mixtures," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 4, pp. 800–814, 2019.

[4] E. Indenbom, N. C. Ristea, A. Saabas, T. Pärnamaa, J. Gužvin, and R. Cutler, "DeepVQE: Real Time Deep Voice Quality Enhancement for Joint Acoustic Echo Cancellation, Noise Suppression and Dereverberation," in *Proc. INTERSPEECH 2023*, 2023, pp. 3819–3823.

[5] H. Dubey, V. Gopal, R. Cutler, A. Aazami, S. Matusevych, S. Braun, S. E. Eskimez, M. Thakker, T. Yoshioka, H. Gamper *et al.*, "ICASSP 2022 Deep Noise Suppression Challenge," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 9271–9275.

[6] H. Dubey, A. Aazami, V. Gopal, B. Naderi, S. Braun, R. Cutler, A. Ju, M. Zohourian, M. Tang, H. Gamper *et al.*, "ICASSP 2023 Deep Noise Suppression Challenge," *arXiv preprint arXiv:2303.11510*, 2023.

[7] Y. Ju, W. Rao, X. Yan, Y. Fu, S. Lv, L. Cheng, Y. Wang, L. Xie, and S. Shang, "TEA-PSE: Tencent-ethereal-audio-lab personalized speech enhancement system for ICASSP 2022 DNS Challenge," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 9291–9295.

[8] Y. Ju, J. Chen, S. Zhang, S. He, W. Rao, W. Zhu, Y. Wang, T. Yu, and S. Shang, "TEA-PSE 3.0: Tencent-Ethereal-Audio-Lab Personalized Speech Enhancement System For ICASSP 2023 DNS-Challenge," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–2.

[9] X. Yan, Y. Yang, Z. Guo, L. Peng, and L. Xie, "The NPU-Elevoc Personalized Speech Enhancement System for Icassp2023 DNS Challenge," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–2.

[10] J. Yu, H. Chen, Y. Luo, R. Gu, W. Li, and C. Weng, "TSpeech-AI System Description to the 5th Deep Noise Suppression (DNS) Challenge," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.

[11] R. Cutler, A. Saabas, T. Pärnamaa, M. Purin, E. Indenbom, N.-C. Ristea, J. Gužvin, H. Gamper, S. Braun, and R. Aichner, "ICASSP 2023 Acoustic Echo Cancellation Challenge," 2023.

[12] Z. Chen, X. Xia, S. Sun, Z. Wang, C. Chen, G. Xie, P. Zhang, and Y. Xiao, "A progressive neural network for acoustic echo cancellation," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–2.

[13] R. Giri, S. Venkataramani, J.-M. Valin, U. Isik, and A. Krishnaswamy, "Personalized PercepNet: Real-Time, Low-Complexity Target Voice Separation and Enhancement," in *Proc. Interspeech 2021*, 2021, pp. 1124–1128.

[14] C. Xu, W. Rao, E. S. Chng, and H. Li, "Spex: Multi-scale time domain speaker extraction network," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 28, pp. 1370–1384, 2020.

[15] X. Liu, X. Li, and J. Serrà, "Quantitative evidence on overlooked aspects of enrollment speaker embeddings for target speaker separation," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[16] A. Sivaraman and M. Kim, "Zero-shot personalized speech enhancement through speaker-informed model selection," in *2021 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2021, pp. 171–175.

[17] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[18] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1314–1324.

[19] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[20] S. E. Eskimez, T. Yoshioka, H. Wang, X. Wang, Z. Chen, and X. Huang, "Personalized speech enhancement: New models and comprehensive evaluation," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 356–360.

[21] G. Zhang, L. Yu, C. Wang, and J. Wei, "Multi-scale temporal frequency convolutional network with axial attention for speech enhancement," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 9122–9126.

[22] T. Zhou, Y. Zhao, and J. Wu, "ResNeXt and Res2Net structures for speaker verification," in *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2021, pp. 301–307.

[23] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep speaker recognition," 2018.

[24] Z. Wang, R. Giri, D. Shah, J.-M. Valin, M. M. Goodwin, and P. Smaragdis, "A framework for unified real-time personalized and non-personalized speech enhancement," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[25] J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal *et al.*, "The AMI meeting corpus: A pre-announcement," in *International workshop on machine learning for multimodal interaction*. Springer, 2005, pp. 28–39.

[26] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)," in *Proceedings of ICLR*, 2016.

[27] S. Braun and I. Tashev, "A consolidated view of loss functions for supervised deep learning-based speech enhancement," in *2021 44th International Conference on Telecommunications and Signal Processing (TSP)*. IEEE, 2021, pp. 72–76.

[28] C. K. Reddy, V. Gopal, and R. Cutler, "DNSMOS P.835: A non-intrusive perceptual objective speech quality metric to evaluate noise suppressors," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 886–890.

[29] M. Purin, S. Sootla, M. Sponza, A. Saabas, and R. Cutler, "AEC-MOS: A speech quality assessment metric for echo impairment," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 901–905.

[30] 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 *2001 IEEE international conference on acoustics, speech, and signal processing. Proceedings (Cat. No. 01CH37221)*, vol. 2. IEEE, 2001, pp. 749–752.

[31] B. Naderi and R. Cutler, "Subjective Evaluation of Noise Suppression Algorithms in Crowdsourcing," in *INTERSPEECH*, 2021.