PERSONALIZED LOW-BITRATE CODED SPEECH RESTORATION VIA EMBEDDING CONDITIONING AND CLUSTER-BASED MODEL SELECTION

Eray Özgünay[1,2,3], *Christian Rollwage*[1], *Jan Rennies*[1,2]
Xavier Domont[3], *Mattes Ohlenbusch*[1], *Simon Doclo*[1,2]

[1]Fraunhofer Institute for Digital Media Technology IDMT, Oldenburg Branch for Hearing, Speech and Audio Technology HSA, Germany
[2]Carl von Ossietzky Universität Oldenburg, Dept. of Medical Physics and Acoustics, Germany
[3]CEOTRONICS AG, Germany

ABSTRACT

In this paper, we investigate personalization for low-bitrate coded speech restoration, using a non-personalized HiFiGAN+ model as the baseline. We propose two personalization methods based on speaker embeddings, which incur little additional computational cost at inference: (i) embedding conditioning of a single HiFiGAN+ model, and (ii) cluster-based model selection, where multiple personalized models are trained on clusters of speakers. Experiments are performed on AMR-NB coded speech at 4.75 kbps, restoring 8 kHz coded speech to 16 kHz wideband speech. Experimental results with disjoint seen- and unseen-speaker test sets show that cluster-based model selection consistently achieves better objective performance metrics than the non-personalized baseline, while embedding conditioning does not consistently improve performance.

Index Terms— personalization, coded speech restoration, bandwidth extension, AMR-NB, HiFiGAN+, speaker embeddings

1. INTRODUCTION

Despite recent advances in neural speech and audio coding [1], legacy low-bitrate codecs such as AMR-NB [2] remain widely used in low-resource telephony, where they restrict the coded speech bandwidth, degrade listening quality, and contribute to listener fatigue [3]. Therefore, bandwidth extension (BWE) is typically included as part of coded speech restoration (CSR) [4–6]. However, BWE can also magnify existing artefacts: distortions that are inaudible in the lower band may be shifted into higher frequencies, potentially reducing overall quality [6]. Therefore, a target sampling rate of 16 kHz is often an appropriate compromise between increasing speech bandwidth beyond narrowband telephony and avoiding excessive amplification of codec artefacts at higher target sampling rates. Nevertheless, even at 16 kHz, residual codec artefacts and a loss of speaker-specific nuances may persist, raising the question of whether personalized CSR can help mitigate these effects.

Personalization for CSR is motivated by its success in related areas such as speech enhancement [7–11], BWE [12], text-to-speech synthesis [13], automatic speech recognition [14, 15], voice activity detection and diarization [16, 17], and neural speech codecs [18]. In these works, two main methods are commonly explored. One line of research conditions a single model on speaker embeddings [10–17], allowing a single network to adapt to different speakers. Another line trains multiple models for small groups (clusters) of speakers, often combined with speaker-informed model selection at test time [7–9, 18]. Such clusters are typically obtained by grouping speakers in the embedding space so that each cluster contains speakers with similar embeddings. A personalized system can, in principle, be better adapted to a specific speaker by exploiting characteristic voice traits such as rhythm and timbre [19]. However, to the best of our knowledge, personalization has not yet been systematically investigated for CSR.

In this work, we investigate personalization for CSR using HiFiGAN+ [20] as a non-personalized CSR baseline, aiming to restore 8 kHz AMR-NB coded speech at 4.75 kbps to 16 kHz wideband speech. Although our experiments target AMR-NB as a representative of legacy low-bitrate codecs, the proposed methods only require access to coded and clean references during training and are therefore, in principle, applicable to other codecs. At test time, we assume that, for each coded test utterance, a different short clean enrollment utterance from the same speaker is available, from which we compute a speaker embedding. We propose and compare two alternative personalization models for CSR: (a) embedding conditioning, which conditions a single HiFiGAN+ model on a speaker embedding; and (b) cluster-based model selection, which trains one HiFiGAN+ model per cluster. Both methods are designed to introduce only minimal additional computational cost in the CSR model at test time. A systematic comparison is provided on the VCTK dataset [21] using three objective performance metrics (Log-Spectral Distance (LSD) [22], WB-PESQ [23], ViSQOL [24]) for both seen speakers included in the training set and unseen speakers that were held out.

2. METHOD

This section describes the HiFiGAN+ CSR baseline (Sec. 2.1) and both proposed personalization methods: embedding conditioning (Sec. 2.3) and cluster-based model selection (Sec. 2.4).

2.1. CSR Baseline

Let u denote a 16 kHz clean reference utterance and x the corresponding AMR-NB coded speech signal. A CSR system aims to reconstruct the clean utterance u from the coded input x, yielding a restored 16 kHz estimate $\hat{u}$.

The CSR baseline is based on HiFiGAN+ [20], which consists of a non-causal WaveNet [25] generator, waveform and spectrogram discriminators, and a combination of adversarial and multi-resolution reconstruction losses. In this paper, we use the same generator architecture as in [20]: the WaveNet generator uses 2 resid-

This project is funded by the European Union under HorizonEU research and innovation program EASYLI (101119297).

ual stacks with 8 dilated layers per stack, 128 channels, kernel size 3, and a dilation base of 3. We also adopt the same discriminator types (one mel-spectrogram discriminator and multi-scale waveform discriminators) and the same combination of loss terms as in HiFi-GAN+, but adapt their sampling rates, convolution kernel sizes, and STFT/mel parameters to the 16 kHz setting (see Sec. 3.2 for details).

2.2. Speaker Embeddings and Enrollment

All personalization methods use speaker embeddings obtained from a pretrained ECAPA-TDNN [26] model for speaker verification, trained on the VoxCeleb [27] dataset and kept frozen in all experiments. The classification head is discarded and only the embedding part is used, yielding fixed-length 192-dimensional vectors. Let $\mathcal{U}$ denote the set of all 16 kHz clean reference utterances, and $\mathcal{X}$ the corresponding set of AMR-NB coded utterances. We write $f : \mathcal{U} \to \mathbb{R}^{192}$ for the speaker embedder, which maps a clean utterance $u \in \mathcal{U}$ to a speaker embedding $\mathbf{e} = f(u)$.

Training-time embeddings: For each speaker s in the training set, let $\mathcal{U}_s^{\text{train}} \subset \mathcal{U}$ denote the set of clean utterances that are used as training targets. For each speaker s, a single training-time embedding is computed by averaging, i.e.

$$\mathbf{e}_s^{\text{train}} = \frac{1}{|\mathcal{U}_s^{\text{train}}|} \sum_{u \in \mathcal{U}_s^{\text{train}}} f(u). \tag{1}$$

The training-time embedding $\mathbf{e}_s^{\text{train}}$ is used both for embedding conditioning during training (Sec. 2.3) and for defining the clusters in the cluster-based model selection method (Sec. 2.4). All training-time embeddings are normalized to unit norm ($\|\mathbf{e}_s^{\text{train}}\|_2 = 1$).

Test-time embeddings: For each speaker s in the test set, let $\mathcal{U}_s^{\text{test}} \subset \mathcal{U}$ denote the set of clean utterances that are used as test targets, and let $\mathcal{X}_s^{\text{test}} \subset \mathcal{X}$ be the corresponding coded utterances. We index the utterances of speaker s by i. The i-th coded test utterance and its clean target are denoted by $x_{s,i}^{\text{test}} \in \mathcal{X}_s^{\text{test}}$ and $u_{s,i}^{\text{test}} \in \mathcal{U}_s^{\text{test}}$, respectively. For each target utterance $u_{s,i}^{\text{test}}$, we randomly select an enrollment utterance $u_{s,j}^{\text{test}} \in \mathcal{U}_s^{\text{test}}$ from the same speaker with $j \neq i$; it should be noted that the same enrollment utterance $u_{s,j}^{\text{test}}$ may be used for multiple test utterances of the same speaker. The test-time embedding used for $u_{s,i}^{\text{test}}$ is defined as $\mathbf{e}_{s,j}^{\text{test}} = f(u_{s,j}^{\text{test}})$ and is normalized to unit norm. In contrast to the training-time embedding, which is a single per-speaker average, each test-time embedding is computed from one enrollment utterance. These test-time embeddings are used both for embedding conditioning at test time (Sec. 2.3) and to select the cluster in the cluster-based model selection method (Sec. 2.4).

For notational simplicity, $\mathbf{e}$ is used to denote a generic embedding when the distinction between training- and test-time embeddings is not needed; otherwise, we explicitly write $\mathbf{e}_s^{\text{train}}$ or $\mathbf{e}_{s,j}^{\text{test}}$. In all cases, these embeddings are ℓ_2-normalized as defined above.

2.3. Speaker Embedding Conditioning

The first personalization method for CSR conditions a single HiFi-GAN+ CSR model on a speaker embedding, inspired by the global conditioning scheme of WaveNet [25]. A high-level view of this method is shown in Fig. 1. The generator and discriminators are identical to the CSR baseline in Sec. 2.1, except that each residual block in the generator is conditioned on a speaker embedding. As illustrated in Fig. 2, each residual block follows a gated structure: the input is processed by convolutions that produce intermediate feature maps for the *filter* and *gate* paths, denoted by $\mathbf{F}$ and $\mathbf{G}$, respectively.

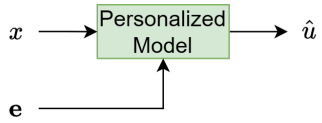

Fig. 1. High-level view of speaker embedding conditioning during training and test: the personalized model maps the coded utterance x and the embedding $\mathbf{e}$ (training- or test-time embedding as defined in Sec. 2.2) to the restored utterance $\hat{u}$.

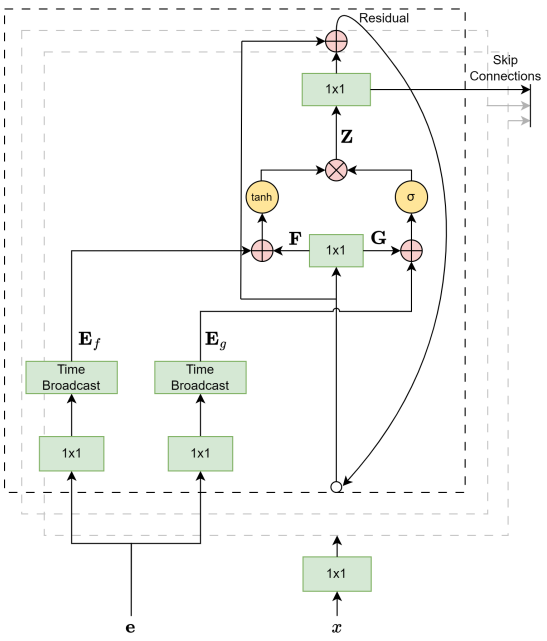

Fig. 2. Low-level implementation of speaker embedding conditioning inside a gated residual block of the HiFiGAN+ generator, used in both training and test. Only a single block is shown; the outgoing arrows denote the residual connection to the next block and the skip connection to the stack output. In each block, the embedding $\mathbf{e}$ (Sec. 2.2) is projected to the channel dimension, broadcast in time, and added to the filter ($\mathbf{F}$) and gate ($\mathbf{G}$) branches before the nonlinearities, yielding the conditioned activation $\mathbf{Z}$.

The 192-dimensional embedding $\mathbf{e} \in \mathbb{R}^{192}$ (Sec. 2.2) is mapped to the internal channel dimension of the generator using two independent learned linear projections (implemented as 1×1 convolutions), one for the filter branch and one for the gate branch. The resulting vectors are broadcast along the temporal axis to obtain feature maps $\mathbf{E}_f$ and $\mathbf{E}_g$ with the same temporal length as $\mathbf{F}$ and $\mathbf{G}$. Speaker conditioning is then performed by adding these feature maps to the corresponding branches before the nonlinearities, and the gated activation is computed as

$$\mathbf{Z} = \tanh\big(\mathbf{F} + \mathbf{E}_f\big) \odot \sigma\big(\mathbf{G} + \mathbf{E}_g\big), \tag{2}$$

where $\odot$ denotes element-wise multiplication and σ is the sigmoid function. The resulting feature map $\mathbf{Z}$ is the output of the gated part of each block and is combined with the residual connection and passed to the next block, so that subsequent layers operate on activations that already incorporate the speaker embedding.

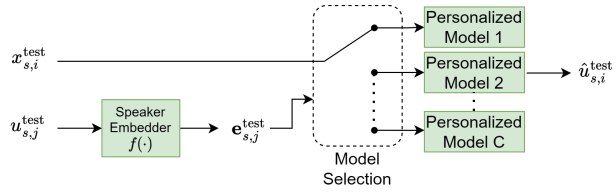

Fig. 3. Cluster-based model selection at test time. For each coded test utterance $x_{s,i}^{\text{test}}$, a different clean enrollment utterance $u_{s,j}^{\text{test}}$ from the same speaker is passed through the speaker embedder to obtain $\mathbf{e}_{s,j}^{\text{test}} = f(u_{s,j}^{\text{test}})$. This embedding is assigned to the nearest cluster centroid, and the corresponding cluster-specific HiFiGAN+ model is then used to generate the restored utterance $\hat{u}_{s,i}^{\text{test}}$ from $x_{s,i}^{\text{test}}$.

2.4. Cluster-based Model Selection

The second personalization method for CSR utilizes multiple HiFi-GAN+ models, each trained on a cluster of speakers that are close to each other in the speaker embedding space. The key idea is that speakers with similar embeddings are expected to share characteristic voice traits; training a separate model on each cluster allows the network to focus on a narrower sub-task confined to a subset of speakers, similarly to cluster-specific neural speech codecs [18].

Training cluster-based models: All training-time (normalized) speaker embeddings $\mathbf{e}_s^{\text{train}}$ in Sec.2.2 are partitioned into C disjoint clusters using k-means in the embedding space. Following [18], we use $C = 4$ in our experiments. With 89 training speakers, k-means yields clusters of 16, 17, 22, and 34 speakers, such that each personalized HiFiGAN+ model is trained on speech from 16–34 speakers. For each cluster, we train a separate HiFiGAN+ CSR model on all utterances from the speakers assigned to that cluster. Apart from the speaker sets used for training, all other aspects of training (losses, optimizer, learning rate schedule, and architectural hyperparameters) are identical to the CSR baseline system described in Sec. 2.1. Thus, each cluster-based model has the same structure and parameter count as the baseline model, with the same computational cost, but is optimized only for a subset of acoustically similar speakers.

Test-time model selection: The cluster-based model selection at test time is illustrated in Fig. 3. For each coded test utterance $x_{s,i}^{\text{test}}$, we use the enrollment embedding $\mathbf{e}_{s,j}^{\text{test}}$ from the same speaker (computed on a different enrollment utterance $u_{s,j}^{\text{test}}$). The embedding $\mathbf{e}_{s,j}^{\text{test}}$ is assigned to the closest cluster centroid in the embedding space (using cosine distance), and the corresponding cluster-specific HiFiGAN+ model is selected to perform CSR.

It should be noted that, in contrast to the speaker embedding conditioning method in Sec. 2.3, the speaker embeddings in this cluster-based method are not used as conditioning inputs to the HiFiGAN+ generator. During training, only the training-time embeddings $\mathbf{e}_s^{\text{train}}$ are used to define the speaker clusters. At test time, the test-time embeddings $\mathbf{e}_{s,j}^{\text{test}}$ are used to select the appropriate cluster-specific model, and all HiFiGAN+ architectures remain identical to the CSR baseline.

3. EXPERIMENTAL SETUP

3.1. Dataset and Splits

Experiments are conducted on the VCTK dataset [21], which comprises 109 English speakers with roughly 400 utterances per speaker. All recordings are resampled to 16 kHz and used as clean reference signals. The clean reference signals are downsampled to 8 kHz and are coded with the AMR-NB codec at 4.75 kbps using `ffmpeg`. The coded signals serve as input to the CSR systems, while the original 16 kHz utterances are used as targets.

The speakers are split into three disjoint groups: 89 speakers for training, 10 speakers for validation, and 10 speakers held out entirely for testing on unseen speakers. For the 89 training speakers, we perform a per-speaker utterance split: for each training speaker, 80% of their utterances are used for training and the remaining 20% are held out for testing. The held-out utterances from these 89 speakers form the seen-speaker test set, while the utterances from the 10 held-out speakers form the unseen-speaker test set.

3.2. Training Configuration

All systems, i.e. the CSR baseline, the embedding conditioning method, and the cluster-based model selection method, share the same architecture and optimization hyperparameters; the embedding conditioning method only adds the 1×1 projection layers. Training is performed on randomly cropped 1 s segments with a batch size of 4. The generator and discriminators follow HiFiGAN+ [20] in terms of the number of layers, strides, channels, and groups. The waveform discriminators operate at four sampling rates $\{2, 4, 8, 16\}$ kHz and use convolution kernel sizes that are linearly scaled with the target sampling rate to preserve the temporal coverage of the original HiFiGAN+ configuration.

We use the HiFiGAN+ loss, combining adversarial losses from waveform and spectrogram discriminators with multi-resolution STFT, log-mel, and time-domain ℓ_1 reconstruction losses. At 16 kHz, the multi-resolution STFT loss uses 4 STFTs whose analysis windows and hop sizes are obtained by linearly scaling the original 48 kHz HiFiGAN+ configuration to 16 kHz, so that the underlying time resolutions remain unchanged. The log-mel loss and the spectrogram discriminators both operate on 128-bin mel spectrograms at 16 kHz, computed using a 25 ms window, 10 ms hop size, and a frequency range of 4–8 kHz, focusing on the reconstructed high band.

We use the Adam optimizer [28] for both the generator and the discriminators with an initial learning rate of 10^{-3} and default hyperparameters $(\beta_1, \beta_2) = (0.9, 0.999)$. Training starts with a 10-epoch warm-up phase in which only the generator is updated using the non-adversarial content losses (i.e. no discriminator updates). After warm-up, adversarial and feature-map losses are enabled and both networks are trained jointly. Following the original HiFiGAN+ schedule, the discriminators are updated at every iteration, while the generator is updated every other iteration (i.e. two discriminator updates per generator update). After warm-up, we apply a PyTorch `LambdaLR` scheduler to the generator optimizer, multiplying its learning rate by 0.01 (i.e., reducing it from 10^{-3} to 10^{-5}), while the discriminator learning rate is kept constant at 10^{-3}. The baseline model and the embedding conditioning model are trained for 20 epochs. Each model in the cluster-based model selection method is trained for the same total number of optimization steps as the 20-epoch baseline model, which, due to the smaller per-cluster dataset size, corresponds to more than 20 epochs for each cluster-specific model.

Validation is performed every fixed number of optimization steps using the same CSR pipeline as at test time. For the baseline and the embedding conditioning method, this corresponds to one validation every 3 epochs. Early stopping is based on ViSQOL, WB-PESQ, and LSD on the validation set: after each validation, if at least one of the three metrics improves over its current best value, we update the best checkpoint and reset a patience counter. If

none of the three metrics improves for three consecutive validation steps, training is stopped and the best checkpoint observed so far is retained for testing. For the cluster-based models, we keep the same validation frequency in terms of optimization steps as the 20-epoch baseline model, which corresponds to more frequent validations than every 3 epochs due to the smaller per-cluster dataset size.

3.3. Performance Metrics

Testing is performed on both seen- and unseen-speaker sets by comparing the restored 16 kHz output signals with their corresponding 16 kHz clean reference signals. We consider three intrusive objective performance metrics: LSD, which quantifies spectral distortion (lower is better); WB-PESQ, computed using the Python `pesq` package (version 0.0.4); and ViSQOL (version 3.3.3) in speech mode at 16 kHz. WB-PESQ and ViSQOL are perceptual similarity measures between reference and degraded speech (higher is better). For both seen and unseen conditions, and each CSR system, we report the mean and standard deviation of each metric across all test utterances.

3.4. Model Size and Complexity

To characterize the computational footprint of the proposed methods (excluding the ECAPA-TDNN embedder), we report the number of trainable parameters and the multiply–accumulate operations (MACs) per 1 s segment of 8 kHz narrowband input (8,000 samples). The CSR baseline has 1.061 M parameters and requires 30.54 GMACs. The embedding conditioning method introduces two additional 1×1 convolutional layers in each residual block, slightly increasing the parameter count to 1.459 M and the complexity by 0.0004 GMACs (corresponding to about 0.001% additional computations). For the cluster-based model selection, each cluster-specific model shares the same architecture and complexity as the baseline (1.061 M parameters, 30.54 GMACs). With $C = 4$ clusters, the total parameter count is therefore 4.244 M parameters. At run time, however, only one of these models is active for a given utterance, so the computational complexity remains identical to the baseline, at the expense of storing C separate models.

4. RESULTS

For all methods considered, results for the seen- and unseen-speaker conditions are shown in Tables 1 and 2, respectively, where boldface indicates the best score for each metric. All improvements and differences discussed in this section are based solely on these objective performance metrics. For both conditions, cluster-based model selection achieves lower LSD score values and higher WB-PESQ and ViSQOL score values than the baseline. For embedding conditioning, the LSD and WB-PESQ scores remain very close to the baseline for both conditions, while the ViSQOL score increases. Hence, cluster-based model selection provides the strongest overall performance in terms of all objective metrics, whereas embedding conditioning offers a single-model alternative, only improving ViSQOL.

On average, ViSQOL scores are higher for unseen than for seen speakers for all systems, including the baseline. As ViSQOL is sensitive to the specific speech material [24], we hypothesize that this may mainly reflect differences in the speakers and utterances included in the seen and unseen test sets rather than genuinely better performance on unseen speakers.

Table 1. Results for seen speakers (mean $\pm$ std).

Method	LSD	ViSQOL	WB-PESQ
CSR baseline	0.95 ± 0.03	2.76 ± 0.49	2.21 ± 0.32
Embedding conditioning	0.95 ± 0.03	2.95 ± 0.51	2.22 ± 0.31
Cluster-based model selection	$\mathbf{0.92 \pm 0.03}$	$\mathbf{3.00 \pm 0.48}$	$\mathbf{2.39 \pm 0.33}$

Table 2. Results for unseen speakers (mean $\pm$ std).

Method	LSD	ViSQOL	WB-PESQ
CSR baseline	0.94 ± 0.03	2.90 ± 0.53	2.18 ± 0.34
Embedding conditioning	0.95 ± 0.03	3.01 ± 0.52	2.15 ± 0.32
Cluster-based model selection	$\mathbf{0.92 \pm 0.03}$	$\mathbf{3.06 \pm 0.47}$	$\mathbf{2.34 \pm 0.33}$

5. CONCLUSION

This paper investigated personalization for low-bitrate coded speech restoration on the legacy AMR-NB codec at 4.75 kbps using a HiFi-GAN+ CSR baseline and ECAPA-TDNN speaker embeddings. Two personalization methods were considered: embedding conditioning of a single model and cluster-based model selection over multiple models trained on speaker clusters. Results on the VCTK dataset showed that embedding conditioning only improved ViSQOL scores over the baseline, while increasing the model size from 1.061 M to 1.459 M parameters and hardly increasing computational complexity. Results also showed that cluster-based model selection consistently improved all objective performance metrics over the baseline for both seen and unseen speaker conditions, indicating that using speaker information for clustering can benefit CSR. Moreover, cluster-based model selection keeps the per-model complexity equal to the baseline but requires storing $C = 4$ times as many parameters.

Future work includes subjective listening tests to assess whether the objective improvements correspond to perceived quality differences. We will also investigate additional personalization strategies, including hybrids that combine embedding conditioning and cluster-based model selection, and systematically analyse design choices such as the number and size of clusters. Finally, we plan to investigate personalization in lightweight, real-time CSR models for low-resource deployment, focusing on the trade-off between objective performance, model size, and computational complexity.

6. REFERENCES

[1] M. Kim and J. Skoglund, "Neural Speech and Audio Coding: Modern AI technology meets traditional codecs," *IEEE Signal Processing Magazine*, vol. 41, no. 6, pp. 85–93, 2024.

[2] 3rd Generation Partnership Project (3GPP), "Mandatory Speech Codec Speech Processing Functions; Adaptive Multi-Rate (AMR) Speech Codec; Transcoding Functions," Tech. Rep. TS 26.071, 3rd Generation Partnership Project (3GPP), 1999.

[3] J. Büthe and J.-M. Valin, "A Lightweight and Robust Method for Blind Wideband-to-Fullband Extension of Speech," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, Tahoe City, CA, USA, 2025.

[4] K. Schmidt, B. Edler, A. M. Mahmoud, and G. Fuchs, "LPC-GAN for Speech Super-Resolution," in *Proc. European Signal Processing Conference (EUSIPCO)*, Helsinki, Finland, Sept. 2023, pp. 346–350.

[5] L. Wen, L. Wang, Y. Zheng, W. Shi, and K. P. Choi, "FT-CSR: Cascaded Frequency-Time Method for Coded Speech Restoration," in *Proc. IEEE International Conference on Multimedia and Expo (ICME)*, Niagara Falls, ON, Canada, July 2024.

[6] K. Gupta, S. Korse, A. Brendel, N. Pia, and G. Fuchs, "UBGAN: Enhancing Coded Speech With Blind and Guided Bandwidth Extension,"

in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, Tahoe City, CA, USA, 2025.

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

[8] A. Sivaraman and M. Kim, "Sparse Mixture of Local Experts for Efficient Speech Enhancement," in *Proc. Interspeech*, Shanghai, China, Oct. 2020, pp. 4526–4530.

[9] A. Sivaraman and M. Kim, "Zero-Shot Personalized Speech Enhancement Through Speaker-Informed Model Selection," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY, USA, Oct. 2021, pp. 171–175.

[10] E. Eskimez, T. Yoshioka, H. Wang, X. Wang, Z. Chen, and X. Huang, "Personalized Speech Enhancement: New Models and Comprehensive Evaluation," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Singapore, May 2022, pp. 356–360.

[11] K. Žmolíková, M. Delcroix, T. Ochiai, K. Kinoshita, J. Černockỳ, and D. Yu, "Neural Target Speech Extraction: An Overview," *IEEE Signal Process. Mag.*, vol. 40, no. 3, pp. 8–29, May 2023.

[12] P. Xu, Z. Zhang, and Z. Fu, "Personalized Bone-Conduction Bandwidth Extension With Speaker Characteristics," in *Proc. Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Shangri-la, Singapore, Oct. 2025, pp. 1128–1133.

[13] E. Cooper, C.-I. Lai, Y. Yasuda, F. Fang, X. Wang, N. Chen, and J. Yamagishi, "Zero-Shot Multi-Speaker Text-to-Speech With State-of-the-Art Neural Speaker Embeddings," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, May 2020, pp. 6184–6188.

[14] M. Delcroix, K. Žmolíková, K. Kinoshita, A. Ogawa, and T. Nakatani, "Single-Channel Target Speaker Extraction and Recognition With SpeakerBeam," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, AB, Canada, Apr. 2018, pp. 5554–5558.

[15] P. Denisov and T. Vu, "End-to-End Multi-Speaker Speech Recognition Using Speaker Embeddings and Transfer Learning," in *Proc. Interspeech*, Graz, Austria, Sept. 2019, pp. 4425–4429.

[16] S. Ding, Q. Wang, S.-y. Chang, L. Wan, and I. L. Moreno, "Personal VAD: Speaker-Conditioned Voice Activity Detection," in *Proc. Odyssey 2020: The Speaker and Language Recognition Workshop*, Tokyo, Japan, Nov. 2020, pp. 433–439.

[17] I. Medennikov, M. Korenevsky, T. Prisyach, Y. Khokhlov, M. Korenevskaya, I. Sorokin, T. Timofeeva, A. Mitrofanov, A. Andrusenko, I. Podluzhny, et al., "Target-Speaker Voice Activity Detection: A Novel Approach for Multi-Speaker Diarization in a Dinner Party Scenario," in *Proc. Interspeech*, Shanghai, China, Oct. 2020, pp. 274–278.

[18] I. Jang, H. Yang, W. Lim, S. Beack, and M. Kim, "Personalized Neural Speech Codec," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Seoul, South Korea, Apr. 2024, pp. 991–995.

[19] W. Wang, Y. Song, and S. Jha, "USAT: A Universal Speaker-Adaptive Text-to-Speech Approach," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 32, pp. 2590–2604, 2024.

[20] J. Su, Y. Wang, A. Finkelstein, and Z. Jin, "Bandwidth Extension Is All You Need," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toronto, ON, Canada, June 2021, pp. 696–700.

[21] J. Yamagishi, C. Veaux, and K. MacDonald, "CSTR VCTK Corpus: English Multi-Speaker Corpus for CSTR Voice Cloning Toolkit (Version 0.92)," Centre for Speech Technology Research (CSTR), University of Edinburgh, 2019.

[22] A. Gray and J. Markel, "Distance measures for speech processing," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 5, pp. 380–391, 2003.

[23] ITU-T, "Wideband Extension to Recommendation P.862 for the Assessment of Wideband Telephone Networks and Speech Codecs," ITU-T Rec. P.862.2, International Telecommunication Union, Geneva, Switzerland, 2005.

[24] M. Chinen, F. S. C. Lim, J. Skoglund, N. Gureev, F. O'Gorman, and A. Hines, "ViSQOL v3: An Open Source Production Ready Objective Speech and Audio Metric," in *Proc. Quality of Multimedia Experience (QoMEX)*, Athlone, Ireland, Oct. 2020.

[25] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukçuoğlu, "WaveNet: A Generative Model for Raw Audio," in *Proc. ISCA Speech Synthesis Workshop (SSW 9)*, Sunnyvale, CA, USA, Sept. 2016, p. 125.

[26] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN-Based Speaker Verification," in *Proc. Interspeech*, Shanghai, China, Oct. 2020, pp. 3830–3834.

[27] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: A Large-Scale Speaker Identification Dataset," in *Proc. Interspeech*, Stockholm, Sweden, Aug. 2017, p. 2616.

[28] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proc. 3rd Int. Conf. Learn. Representations (ICLR)*, San Diego, CA, USA, May 2015.