

# AdaptVC: High Quality Voice Conversion with Adaptive Learning

Jaehun Kim<sup>1</sup>, Ji-Hoon Kim<sup>1</sup>, Yeunju Choi<sup>2</sup>, Tan Dat Nguyen<sup>1</sup>, Seongkyu Mun<sup>2</sup>, Joon Son Chung<sup>1</sup>

<sup>1</sup>Korea Advanced Institute of Science and Technology, South Korea, <sup>2</sup>Samsung Research, South Korea  
{kjaehun, jh.kim, tandat.kaist, joonson}@kaist.ac.kr, {wkadldppdy, skmoon777}@gmail.com

**Abstract**—The goal of voice conversion is to transform the speech of a source speaker to sound like that of a reference speaker while preserving the original content. A key challenge is to extract disentangled linguistic content from the source and voice style from the reference. While existing approaches leverage various methods to isolate the two, a generalization still requires further attention, especially for robustness in zero-shot scenarios. In this paper, we achieve successful disentanglement of content and speaker features by tuning self-supervised speech features with adapters. The adapters are trained to dynamically encode nuanced features from rich self-supervised features, and the decoder fuses them to produce speech that accurately resembles the reference with minimal loss of content. Moreover, we leverage a conditional flow matching decoder with cross-attention speaker conditioning to further boost the synthesis quality and efficiency. Subjective and objective evaluations in a zero-shot scenario demonstrate that the proposed method outperforms existing models in speech quality and similarity to the reference speech.

**Index Terms**—self-supervised learning, speech synthesis, voice conversion

## I. INTRODUCTION

Voice Conversion (VC) converts a source speaker’s voice to sound as if it were uttered by a target speaker, preserving the original linguistic content. A powerful voice conversion framework has potential applications, including personalized text-to-speech, privacy security, and language learning tools [1]–[3]. The quest to closely resemble the target speaker’s timbre without losing the original content calls for a successful disentanglement of linguistic and speaker attributes, as well as the generation of rich acoustic representation by effectively fusing the two. Such a capability has much stronger influence in a zero-shot VC scenario, where the source and target voices are completely unseen during training.

Early VC systems have attempted to resolve disentanglement with various techniques. AutoVC [4] constructs an autoencoder architecture with an information bottleneck layer to encode content features only, and F0-AutoVC [5] extends the idea and utilizes the fundamental frequency to improve the quality of generation. DiffVC [6] adopts diffusion mechanism into VC and proposes maximum likelihood sampling that generalizes one-shot VC scenario. However, the models either fail to generate speech with close resemblance to the reference speaker or lose content information and naturalness.

Recently, self-supervised learning (SSL) has drawn increased attention for the utilization of large-scale unlabeled data [7]–[10]. Moreover, the features extracted from a pre-trained SSL model show a high correlation with both the

acoustic and linguistic information [11], suggesting high potential for application in VC research. NANSY [11] utilizes intermediate features extracted with XLS-R [10] as a content representation disentangled from speaker attributes. DDDM-VC [12] follows the similar approach and proposes a dual-path diffusion decoder that separately models source and filter information. kNN-VC [13] proposes a non-parametric approach and replaces the features extracted from source speech with WavLM [9] with the nearest neighbors of those from target speech. Current SSL-based VC systems have brought the conversion quality close to human, but meticulous parameter searches, including heuristic selection of intermediate layers, and the requirement for large computing time remains unsolved.

To address this, this paper presents AdaptVC, a high-quality voice conversion model with nuanced self-supervised speech representations. Inspired by the concept of tuning large-scale pretrained models with the small addition of parameters [14]–[16], the model incorporates adapters that tunes the rich representation from an SSL model and generates nuanced features. Specifically, all intermediate layer outputs of an SSL model are combined via weighted summation, and auxiliary modules based on specific objectives automatically guides the model to produce richer representation than a single layer output. Moreover, high speech quality and fast processing time are achieved through a Conditional Flow Matching decoder with an Optimal Transport objective (OT-CFM) [17] and cross-attention speaker conditioning. The design allows the decoder to effectively model detailed speaker characteristics by offering multiple conditioning operation. Both subjective and objective metrics in a challenging zero-shot scenario demonstrate that AdaptVC surpasses all existing voice conversion models in terms of intelligibility and target speaker similarity. Audio samples are available in the demo page: <https://mm.kaist.ac.kr/projects/AdaptVC>

## II. METHOD

AdaptVC exhibits an encoder-decoder architecture, as illustrated in Fig. 1. Source and reference utterances are fed to separate encoders comprising HuBERT [7], a pretrained speech SSL model, with an adapter to combine all intermediate layer outputs. As illustrated in Fig. 2(a), adapters in the encoders contain learnable weights that serve as coefficients for the weighted summation, and the values are updated to maximize the extraction of content and speaker-only information. The

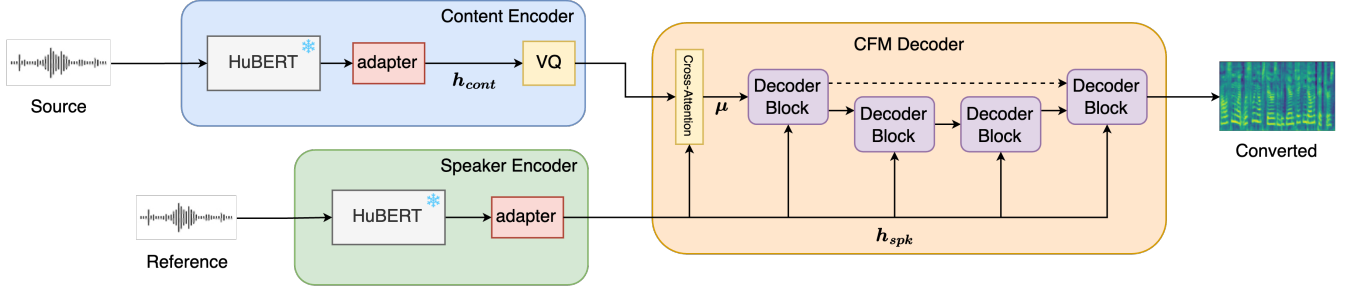


Fig. 1. Overall architecture of AdaptVC.  $h_{cont}$  denotes the content representation from the adapter in the content encoder, and  $h_{spk}$  denotes the speaker features from that in the speaker encoder. Prior distribution  $\mu$  is obtained by fusing the content and speaker information through cross-attention.

encoded content features are passed to a U-Net based CFM decoder, conditioned with encoded speaker features, which generates the mel-spectrogram of the converted speech.

#### A. Content Encoder

The content encoder aims to extract linguistic features and minimize the influence of speaker-specific attributes. However, as the objective of training with non-parallel VC is to reconstruct the original speech, the content encoder naturally tends to produce features rich in both content and speaker information. To further guide the model in disentangling the speaker aspect, a vector quantization (VQ) layer is applied after the adapter. As demonstrated in [18], [19], the quantization of latent features produces discrete while compact representation. From the perspective of speech encoding, the output of the adapter is guided to map similar content information from various speakers into closest embedding, ultimately generating accurate linguistic information independent of speakers.

#### B. Speaker Encoder

The objective of the speaker encoder is to produce rich speaker features independent of linguistic content. Unlike conventional approaches where a single vector of speaker information is utilized as a condition, the model leverages frame-wise speaker features to capture the time-varying timbre of different utterances, as [20], [21] demonstrate speech synthesis with close similarity to the target speaker. A reference speech utterance is passed to the HuBERT model, and the final representation produced by the adapter is fed to the decoder to transform content only features into rich acoustic features.

#### C. CFM Decoder

The decoder receives content and speaker features and generates a mel-spectrogram of the converted speech. To capture efficiency and quality of generation, the model leverages OT-CFM objective [17], [22]. By regressing the transformation to match the mapping between the data and target distribution, OT-CFM provides higher efficiency and robustness compared to a diffusion mechanism, which models a stochastic transformation of data. The decoder is designed to provide speaker

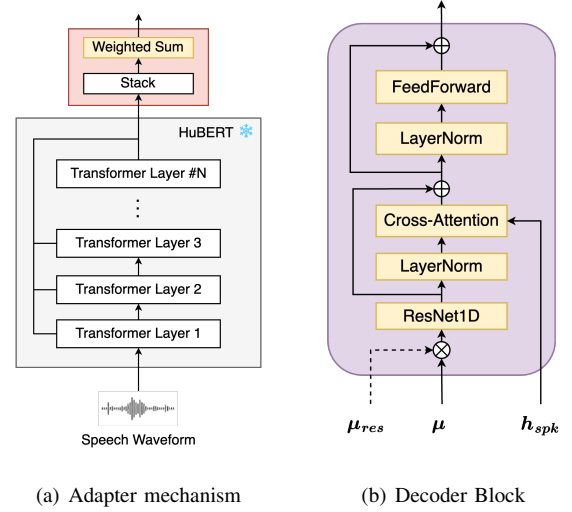


Fig. 2. Illustration of HuBERT adapter mechanism (a) and decoder block in the CFM decoder (b). Residual input  $\mu_{res}$  is concatenated to  $\mu$  only for blocks with skip connections.

condition in multiple ways based on a transformer-based U-Net architecture [22]. Inspired by an effective conditioning method in the image domain [23], self-attention layers in the transformer blocks are replaced with cross-attention layers, where the encoded speaker features serve as keys and values as shown in Fig. 2(b). The provision of multiple conditioning via cross-attention allows the decoder to faithfully model the acoustic details of various speakers, and the adapter in the speaker encoder is optimized to produce rich speaker information by combining multiple outputs of HuBERT.

#### D. Training Objective

The model is trained with three objective functions: commitment loss, prior loss, and OT-CFM loss for the decoder. The commitment loss enforces the input of the VQ layer to commit to the codebook vectors, as in [19]. The loss is formulated as:

$$\mathcal{L}_{commit} = MSE(h_{cont}, sg[e]), \quad (1)$$

TABLE I  
QUALITATIVE AND QUANTITATIVE RESULTS WITH 95% CONFIDENCE INTERVALS ( $\uparrow$ : HIGHER IS BETTER,  $\downarrow$ : LOWER IS BETTER).  
THE PARENTHEZIZED NUMBERS INDICATE THE NUMBER OF SAMPLING STEPS.

| Model               | RTF $\downarrow$ | UTMOS $\uparrow$ | MOS-N $\uparrow$                  | MOS-S $\uparrow$                  | WER $\downarrow$                  | CER $\downarrow$                  | SECS $\uparrow$ |
|---------------------|------------------|------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------|
| GT (vocoded)        | N/A              | 4.07             | 4.51 $\pm$ 0.12                   | 4.04 $\pm$ 0.21                   | 2.42 $\pm$ 0.05                   | 1.08 $\pm$ 0.01                   | 0.982           |
| kNN-VC [13]         | 0.15             | 2.87             | 1.96 $\pm$ 0.15                   | 2.34 $\pm$ 0.27                   | 34.6 $\pm$ 7.53                   | 21.8 $\pm$ 3.97                   | 0.752           |
| Diff-VC (30) [6]    | 0.35             | 3.67             | 3.33 $\pm$ 0.14                   | 3.14 $\pm$ 0.20                   | 13.1 $\pm$ 1.51                   | 6.75 $\pm$ 0.88                   | <b>0.828</b>    |
| Diff-VC (10) [6]    | 0.22             | 3.70             | 3.14 $\pm$ 0.16                   | 2.92 $\pm$ 0.20                   | 12.2 $\pm$ 1.49                   | 6.22 $\pm$ 0.87                   | 0.779           |
| DDDM-VC (30) [12]   | 0.30             | 3.43             | 3.48 $\pm$ 0.15                   | 3.27 $\pm$ 0.22                   | 8.32 $\pm$ 2.09                   | 4.43 $\pm$ 1.18                   | 0.819           |
| DDDM-VC (10) [12]   | 0.19             | 3.51             | 3.48 $\pm$ 0.14                   | 3.19 $\pm$ 0.23                   | 6.40 $\pm$ 2.15                   | 3.37 $\pm$ 1.15                   | 0.823           |
| <b>AdaptVC (10)</b> | 0.04             | <b>3.95</b>      | <b>4.13 <math>\pm</math> 0.14</b> | <b>3.52 <math>\pm</math> 0.21</b> | 7.39 $\pm$ 1.06                   | 3.63 $\pm$ 0.58                   | 0.821           |
| <b>AdaptVC (5)</b>  | 0.02             | 3.94             | 3.86 $\pm$ 0.14                   | 3.36 $\pm$ 0.21                   | 6.96 $\pm$ 0.97                   | 3.29 $\pm$ 0.49                   | 0.801           |
| <b>AdaptVC (1)</b>  | <b>0.01</b>      | 3.38             | 1.76 $\pm$ 0.11                   | 2.14 $\pm$ 0.24                   | <b>6.36 <math>\pm</math> 0.85</b> | <b>2.98 <math>\pm</math> 0.42</b> | 0.768           |

where  $MSE$  is the mean squared error,  $\text{sg}[\cdot]$  is a stop-gradient operator,  $\mathbf{h}_{cont}$  is the output of the adapter in the content encoder, and  $\mathbf{e}$  is the codebook vectors in the VQ layer, which become the content features fed to the decoder.

Following [24], the prior loss minimizes the log-likelihood between the prior distribution and the mel-spectrogram:

$$\mathcal{L}_{prior} = -\sum_{i=1}^T \log \varphi(\mathbf{x}_i; \boldsymbol{\mu}_i, I), \quad (2)$$

where  $\mathbf{x}$  denotes the target mel-spectrogram,  $\varphi(\cdot; \boldsymbol{\mu}_i, I)$  is a probability density function of  $\mathcal{N}(\boldsymbol{\mu}_i, I)$ , and  $T$  denotes the temporal length. The prior loss also directs the codebook vectors in the VQ layer to represent discrete but nuanced information.

The loss for the decoder follows [22], which estimates a vector field with linear trajectory via optimal transport (OT):

$$\mathcal{L}_{dec} = \mathbb{E}_{t,q(\mathbf{x}_1),p_0(\mathbf{x}_0)} \|u_t^{\text{OT}}(\phi_t^{\text{OT}}(\mathbf{x}_0)|\mathbf{x}_1) - v_t(\phi_t^{\text{OT}}(\mathbf{x}_0)|\boldsymbol{\mu}, \mathbf{h}_{spk}; \theta)\|^2, \quad (3)$$

where  $\theta$  denotes the network parameters,  $\phi_t^{\text{OT}}(\mathbf{x}) = (1 - (1 - \sigma_{min})t)\mathbf{x}_0 + t\mathbf{x}_1$  represents flow that maps the source and target distribution,  $u_t$  is a known vector field that generates approximate path from prior distribution  $p_0$  to target data distribution  $p_t$ .  $\mathbf{h}_{spk}$  represents continuous speaker features obtained from the speaker encoder, serving as a condition.

Finally, the total training objective is formulated as:

$$\mathcal{L}_{total} = \mathcal{L}_{commit} + \mathcal{L}_{prior} + \mathcal{L}_{dec}. \quad (4)$$

### III. EXPERIMENT

#### A. Data

The model was trained with LibriTTS [25], a multi-speaker text-to-speech dataset, where train-clean-100 and train-clean-360 subsets were split into training, validation, and test sets following [12]. To evaluate zero-shot VC performance, 20 source speakers and 20 target speakers from the VCTK [26] corpus were utilized. Speech waveforms were resampled to 16kHz to match the input configuration of HuBERT. A log-scale mel-spectrogram was generated with window size and

filter size of 1280, hop size of 320, and mel filterbank size of 80, where the temporal resolution was also adjusted to match that of HuBERT.

#### B. Training

The two adapters in the content and speaker encoders were constructed with a single fully-connected layer without bias, followed by a softmax activation to ensure the sum of the probabilities is 1. The VQ layer in the content encoder employed a single quantizer with codebook size of 512. The architecture of the decoder followed that of [22], where self-attention layers in each decoder block were replaced with cross-attention. A HiFi-GAN [27] vocoder was trained with the training subset to correctly evaluate the performance of the proposed model.

#### C. Baselines

The performance of the model was compared with three established VC models: kNN-VC [13]<sup>1</sup>, DDDM-VC [12]<sup>2</sup>, DiffVC [6]<sup>3</sup>. The samples were generated based on the official implementations. Since the reported kNN-VC model is trained with a different dataset, we trained the model with LibriTTS to ensure a fair comparison. The samples were normalized with respect to the root mean square value before evaluation.

#### D. Evaluation Metrics

Both qualitative and quantitative metrics were obtained for comprehensive evaluation. Mean Opinion Score (MOS) was conducted to 20 domain experts with 40 generated samples. Naturalness (MOS-N) was evaluated from the perspective of general speech quality and the intelligibility by presenting the ground-truth text for a reference, and perceptual similarity to the reference speaker (MOS-S) was measured by juxtaposing the parallel target speech. Moreover, an automatic speech quality estimation model (UTMOS) [28] was utilized. Word Error Rate (WER) and Character Error Rate (CER) were calculated by a pretrained ASR model [29]. Speaker Embedding Cosine

<sup>1</sup><https://github.com/bshall/knn-vc>

<sup>2</sup><https://github.com/hayegong0/DDDM-VC>

<sup>3</sup><https://github.com/trinhluanvubk/Diff-VC>

similarity (SECS) between the generated speech and the target ground-truth is obtained by calculating cosine similarity of the speaker embeddings obtained through *Resemblyzer*<sup>4</sup>, a pretrained speaker verification model [30]. Lastly, Real Time Factor (RTF) was measured for speed comparison.

#### IV. RESULTS

##### A. Quantitative Evaluation

The quantitative metrics (WER, CER, RTF) in Table I demonstrate that the proposed method consistently achieves high intelligibility. The proposed model with a single sampling step results in the lowest WER and CER and RTF. Although increasing the number of sampling steps leads to higher error rates, the observed differences are smaller than those in the baseline models, which highlights the robustness of the proposed method. In conclusion, the proposed model with 5 sampling steps shows the best balance between performance and speed.

##### B. Qualitative Evaluation

As shown in Table I, the proposed method outperforms existing approaches in naturalness and similarity MOS by a significant margin. DiffVC achieves the highest SECS due to its direct use of speaker vectors from the same model [30], while the proposed approach attains the highest perceptual similarity, as reflected in the high MOS-S values. Moreover, AdapterVC with only 5 sampling steps clearly outperforms the baseline models with its speed up to 10 times faster. This demonstrates the strong performance of the proposed method as well as its applicability to real-time scenarios.

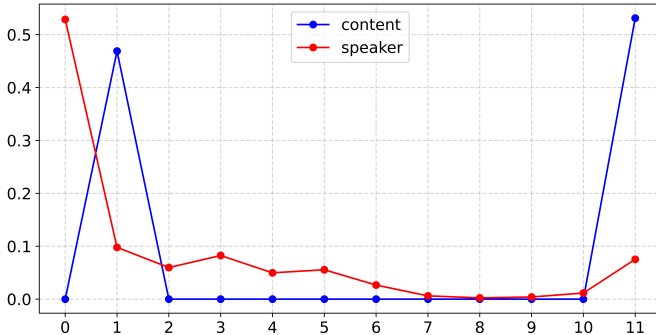


Fig. 3. Visualization of adapter weights. Numbers in the x-axis indicate layer indices and y-axis denotes the trained weights.

##### C. Analysis on Adapter Weights

The weights of the adapters in content and speaker encoders exhibit distinct behaviors, as visualized in Fig. 3. The adapter in the content encoder mainly utilizes the second and last layer output of HuBERT, while the weights of the other layers converge to zero. The adapter in the speaker encoder, on the other hand, shows the highest weight on the first layer and gradually decreases as the layers proceed. This aligns with the

findings in [31], where a speech SSL model shows a tendency to capture acoustic information in the earlier layers and robust linguistic information in the latter, and consequently validates the adaptability of the proposed method.

TABLE II  
QUALITATIVE (UTMOS) AND QUANTITATIVE (WER, SECS) RESULTS OF ABLATION STUDY.

|                              | UTMOS | WER↓        | SECS↑ |
|------------------------------|-------|-------------|-------|
| <b>AdaptVC (5)</b>           | 3.94  | 6.96 ± 0.97 | 0.801 |
| <i>w/o adapters</i>          | 3.81  | 8.47 ± 0.90 | 0.793 |
| <i>w/o VQ</i>                | 3.90  | 1.52 ± 0.36 | 0.648 |
| <i>condition: SALN</i>       | 3.76  | 12.4 ± 1.73 | 0.754 |
| <i>condition: Mean + Add</i> | 3.76  | 12.8 ± 1.90 | 0.756 |

##### D. Ablation Study

To validate the contribution of each module in the proposed method, a systematic ablation study was performed. First, the adapters were replaced with the fixed output – the last layer output for the content encoder, and the first layer for the speaker encoder, as they show high correlation to the linguistic and speaker information [32]. The impact of the VQ layer was evaluated by removing it. The contribution of the cross-attention speaker condition was compared with two established speaker conditioning methods: Style Adaptive Layer Normalization (SALN) [33] and mean pooling followed by its addition to latent content features.

The removal of adapters show a clear decline in the generated speech’s naturalness and intelligibility, as shown in Table II. This indicates that the learned combination of multiple intermediate outputs from HuBERT contains more nuanced information compared to a single layer output. While the model without the VQ layer shows notably high UTMOS and low WER, the similarity of the converted speech to the reference speaker is significantly low. Qualitatively, the model merely reconstructs the source speech regardless of the reference speaker, underscoring the VQ layer’s importance in disentangling content. Finally, a distinct degradation in speaker similarity with the two conditioning methods provides clear evidence that speaker conditioning with multiple cross-attention contributes to accurately resemble the reference speaker.

#### V. CONCLUSION

This paper proposes AdaptVC, a high quality voice conversion model with adaptive learning framework. The proposed adapters automatically determine the optimal combination of intermediate SSL layer outputs, thereby eliminating the need for heuristic parameter tuning and the integration of additional information. The utilization of OT-CFM decoder and speaker conditioning with multiple cross-attention layers efficiently boost the quality of generation. Qualitative and quantitative evaluations suggest that AdaptVC outperforms the existing approaches by a significant margin.

<sup>4</sup><https://github.com/resemble-ai/Resemblyzer>



## REFERENCES

- [1] E. Casanova, J. Weber, C. D. Shulby, A. C. Junior, E. Gölge, and M. A. Ponti, “YourTTS: Towards Zero-Shot Multi-Speaker TTS and Zero-Shot Voice Conversion for Everyone,” in *Proc. ICML*, 2022. 1
- [2] S.-H. Lee, J.-H. Kim, H. Chung, and S.-W. Lee, “VoiceMixer: Adversarial Voice Style Mixup,” in *NeurIPS*, 2021. 1
- [3] H. Lu, D. Wang, X. Wu, Z. Wu, X. Liu, and H. Meng, “Disentangled Speech Representation Learning for One-Shot Cross-Lingual Voice Conversion Using  $\beta$ -VAE,” in *IEEE Spoken Language Technology workshop*, 2023. 1
- [4] K. Qian, Y. Zhang, S. Chang, X. Yang, and M. Hasegawa-Johnson, “AutoVC: Zero-Shot Voice Style Transfer with Only Autoencoder Loss,” in *Proc. ICML*, 2019. 1
- [5] K. Qian, Z. Jin, M. Hasegawa-Johnson, and G. J. Mysore, “F0-Consistent Many-To-Many Non-Parallel Voice Conversion Via Conditional Autoencoder,” in *Proc. ICASSP*, 2020. 1
- [6] V. Popov, I. Vovk, V. Gogoryan, T. Sadekova, M. Kudinov, and J. Wei, “Diffusion-Based Voice Conversion with Fast Maximum Likelihood Sampling Scheme,” in *Proc. ICLR*, 2022. 1, 3
- [7] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhota, R. Salakhutdinov, and A. Mohamed, “HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units,” *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, 2021. 1
- [8] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations,” in *NeurIPS*, 2020. 1
- [9] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao, J. Wu, L. Zhou, S. Ren, Y. Qian, Y. Qian, J. Wu, M. Zeng, X. Yu, and F. Wei, “WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing,” *IEEE Journal of Selected Topics in Signal Processing*, 2022. 1
- [10] A. Babu, C. Wang, A. Tjandra, K. Lakhotia, Q. Xu, N. Goyal, K. Singh, P. von Platen, Y. Saraf, J. M. Pino, A. Baevski, A. Conneau, and M. Auli, “XLS-R: Self-Supervised Cross-Lingual Speech Representation Learning at Scale,” in *Proc. Interspeech*, 2021. 1
- [11] H.-S. Choi, J. Lee, W. Kim, J. Lee, H. Heo, and K. Lee, “Neural Analysis and Synthesis: Reconstructing Speech from Self-Supervised Representations,” in *NeurIPS*, 2021. 1
- [12] H.-Y. Choi, S.-H. Lee, and S.-W. Lee, “DDDM-VC: Decoupled Denoising Diffusion Models with Disentangled Representation and Prior Mixup for Verified Robust Voice Conversion,” in *Proc. AAAI*, 2024. 1, 3
- [13] M. Baas, B. van Niekerk, and H. Kamper, “Voice Conversion With Just Nearest Neighbors,” in *Proc. Interspeech*, 2023. 1, 3
- [14] N. Houlsby, A. Giurui, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, “Parameter-Efficient Transfer Learning for NLP,” in *Proc. ICML*, 2019. 1
- [15] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, “Lora: Low-rank adaptation of large language models,” in *Proc. ICLR*, 2022. 1
- [16] J. Pfeiffer, A. Rücklé, C. Poth, A. Kamath, I. Vulić, S. Ruder, K. Cho, and I. Gurevych, “AdapterHub: A framework for adapting transformers,” in *Proc. EMNLP*, 2020. 1
- [17] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le, “Flow Matching for Generative Modeling,” in *Proc. ICLR*, 2023. 1, 2
- [18] J. Yu, X. Li, J. Y. Koh, H. Zhang, R. Pang, J. Qin, A. Ku, Y. Xu, J. Baldridge, and Y. Wu, “Vector-Quantized Image Modeling with Improved VQGAN,” in *Proc. ICLR*, 2022. 2
- [19] A. Van Den Oord, O. Vinyals *et al.*, “Neural discrete representation learning,” in *NeurIPS*, 2017. 2
- [20] H.-S. Choi, J. Yang, J. Lee, and H. Kim, “NANSY++: Unified Voice Synthesis with Neural Analysis and Synthesis,” in *Proc. ICLR*, 2023. 2
- [21] J. Kong, J. Lee, J. Kim, B. Kim, J. Park, D. Kong, C. Lee, and S. Kim, “Encoding Speaker-Specific Latent Speech Feature for Speech Synthesis,” in *Proc. ICML*, 2024. 2
- [22] S. Mehta, R. Tu, J. Beskow, É. Székely, and G. E. Henter, “Match-TTS: A fast TTS architecture with conditional flow matching,” in *Proc. ICASSP*, 2024. 2, 3
- [23] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” in *Proc. CVPR*, 2022. 2
- [24] V. Popov, I. Vovk, V. Gogoryan, T. Sadekova, and M. Kudinov, “Gratts: A diffusion probabilistic model for text-to-speech,” in *Proc. ICML*, 2021. 3
- [25] H. Zen, V. Dang, R. Clark, Y. Zhang, R. J. Weiss, Y. Jia, Z. Chen, and Y. Wu, “LibriTTS: A Corpus Derived from LibriSpeech for Text-to-Speech,” in *Proc. Interspeech*, 2019. 3
- [26] C. Veaux, J. Yamagishi, K. MacDonald *et al.*, “Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit,” *University of Edinburgh. The Centre for Speech Technology Research (CSTR)*, 2017. 3
- [27] J. Kong, J. Kim, and J. Bae, “HiFi-GAN: Generative Adversarial Networks for Efficient and High Fidelity Speech Synthesis,” in *NeurIPS*, 2020. 3
- [28] T. Saeki, D. Xin, W. Nakata, T. Koriyama, S. Takamichi, and H. Saruwatari, “Utmos: Utokyo-sarulab system for voicemos challenge 2022,” in *Proc. Interspeech*, 2022. 3
- [29] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, “Robust speech recognition via large-scale weak supervision,” in *Proc. ICML*, 2023. 3
- [30] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, “Generalized end-to-end loss for speaker verification,” in *Proc. ICASSP*, 2018. 4
- [31] K. Fujita, H. Sato, T. Ashihara, H. Kanagawa, M. Delcroix, T. Moriya, and Y. Ijima, “Noise-Robust Zero-Shot Text-to-Speech Synthesis Conditioned on Self-Supervised Speech-Representation Model with Adapters,” in *Proc. ICASSP*, 2024. 4
- [32] V. Vielzeuf, “Investigating the Autoencoder Behavior in Speech Self-Supervised Models: a focus on HuBERT’s Pretraining,” *arXiv preprint arXiv:2405.08402*, 2024. 4
- [33] D. Min, D. B. Lee, E. Yang, and S. J. Hwang, “Meta-stylespeech: Multi-speaker adaptive text-to-speech generation,” in *Proc. ICML*, 2021. 4