DVQVC: AN UNSUPERVISED ZERO-SHOT VOICE CONVERSION FRAMEWORK

*Dayong Li, Xian Li, Xiaofei Li**

Westlake University & Westlake Institute for Advanced Study, Hangzhou, China

ABSTRACT

Zero-shot voice conversion (VC) is to convert speech from one speaker to a target speaker while preserving the original linguistic information, given only one reference speech clip of the unseen target speaker. This work proposes a new VC model, and its key idea is to conduct thorough speaker and content disentanglement by adopting an advanced speech encoder plus vector quantization (VQ) as a content encoder, and an advanced speaker encoder for accurate speaker embedding. In addition, we propose a perceptual loss, a speaker constrative loss and an adversarial loss to compensate the content imperfection caused by VQ and to further improve the speech quality/intelligibility. Overall, the proposed model uses only unsupervised features/losses, and achieves excellent VC performance in terms of both speech quality/intelligibility and speaker similarity, for both seen and unseen speakers.

***Index Terms*—** voice conversion, self-supervised learning, vector quantization, zero shot

1. INTRODUCTION

Voice Conversion (VC) is to convert voice from a source speaker's speech to the voice of a target speaker while preserving linguistic information of the source speech. It can be applied in many areas like creating digital human, privacy protection [1] and dysarthria speech conversion [2].

Speech representation disentanglement tries to disentangle speaker information from linguistic information. Once it's achieved, one can only switch speaker information to perform voice conversion. This idea could be done in either supervised or unsupervised manner. Supervised approaches [3, 4] explicitly model linguistic information through ASR (automatic speech recognition), and extract speaker information [5] through a pretrained speaker verification model. Supervised approaches are often more robust and controllable. Especially, one can adjust speed or pitch [6] and insert or remove a part of speech or words as a TTS (text to speech) system does. However, supervised methods normally need a large amount of training speech and corresponding text annotations. Moreover, ASR might filter out too much timbre information that are better be preserved.

* corresponding author

Unsupervised approaches focus on removing speaker information from the speech representation, often by using vector quantization (VQ), such as in VQVC+ [7]. Its subsequent VQMIVC [8] imposes a mutual information constraint to alleviate the leakage of speaker information to the linguistic representation. AVQVC [9] adds a contrastive loss to VQVC+ to to alleviate speaker leakage.

Recently, speech features achieved by self supervised learning (SSL) are exploited to VC. S2VC [10] uses a cross-attention mechanism to several SSL features. S3prl-VC [11] studies the VC applicability of a bunch of SSL speech features, among which vq-wav2vec [12] performs the best in terms of speaker conversion, but at the cost of speech intelligibility loss. It is shown in [13] that SSL features normally contain much speaker information, which means further speaker disentanglement is required on top of the SSL features when they are used for VC.

A successful VC system requires both speaker conversion and content preservation. Speaker conversion can be well conducted as long as speaker leakage is weak and the speaker embedding itself is accurate. However, reducing speaker leakage may harm the content feature. The generalization to unseen speakers is also a key point for zero-shot VC. Inspired by VQVC+ [7], VQGAN [14], s3prl-VC [11] and a recently proposed SSL speaker embedding system loss-gated-learning (LGL) [15], we propose our disentangled VQVC, named DVQVC, in the framework of speaker/content disentanglement. LGL is taken as the speaker encoder to extract accurate speaker embedding. The SSL wav2vec2 representation [16] is taken as the input of a content encoder, as it is strong in representing linguistic information. VQ is also adopted in the content encoder to further remove speaker information, which however harms the linguistic information to an extent. A perceptual loss to the generated speech is then adopted to compensate the loss of linguistic information. Thence, this system will achieve a good balance between speaker conversion and content preservation. In addition, we propose to use a GAN loss to further improve the speech naturalness, a speaker contrastive loss and a VC loss to further improve the speaker generalization capability. Overall, the proposed model uses only unsupervised features/losses, and achieves excellent VC performance in terms of both speech quality/intelligibility and speaker similarity, for both seen and unseen scenarios.

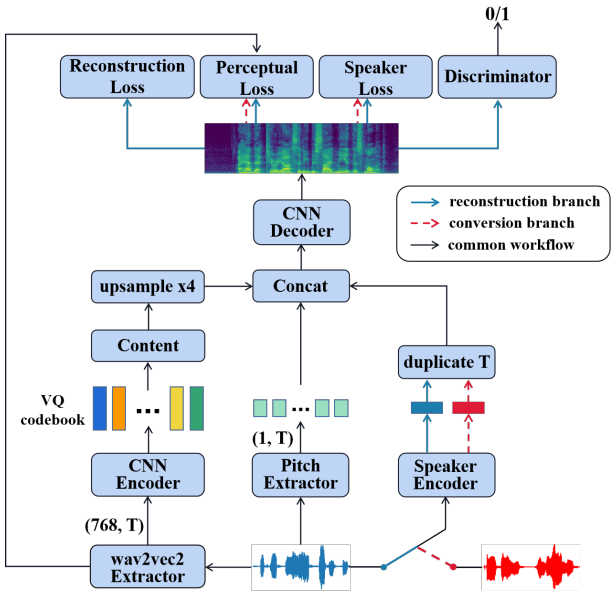

Figure 1. Diagram of the proposed DVQVC system.

2. METHOD

Figure 1 shows the overall architecture of the proposed model, which consists of four modules: 1) a VQ-based content encoder to extract linguistic features from source speech, 2) a pitch extractor to explicitly extract the pitch sequence of source speech, 3) a speaker encoder to extract speaker embedding from either source speech or VC target speech, 4) a decoder to reconstruct mel-spectrogram given the linguistic features, pitch and speaker embedding. During training, an extra discriminator and perceptual block are also used to enhance the reconstructed/converted speech quality.

Speech reconstruction is performed when the speaker embedding of source speech is used, while VC is performed when replacing the speaker embedding of source speech with the one of target speech. In this work, we train the VC model by performing both the speech reconstruction and VC tasks.

2.1. Speech Reconstruction Training

2.1.1. Content encoder and decoder

As s3prl-VC suggests, wav2vec2 [16] shows a great improve in aspect of speech intelligibility for the VC task. In addition, [17] shows that the middle layer's hidden states of wav2vec2 aggregate the most linguistic information. Therefore, we adopt the wav2vec2's hidden states of one middle layer's, denoted as $\mathbf{f} \in \mathbb{R}^{d\times T}$, as the input of our content encoder, where d and T denote the number of hidden units and the sequence length, respectively. In this work, $\mathbf{f}$ is set as the output of the eighth transformer block of wav2vec2, and wav2vec2 is always frozen.

The content encoder transforms $\mathbf{f}$ to a more abstract representation $\mathbf{h} = (h_1, h_2, ..., h_{T/4}) \in \mathbb{R}^{d\times T/4}$, and downsample the sequence length from T to $T/4$. A VQ layer is applied to transform $\mathbf{h}$ to a sequence of discrete codewords $\mathbf{q} = (q_1, q_2, ..., q_{T/4}) \in \mathbb{R}^{d\times T/4}$, where

$$q_j = \underset{q\in\mathbf{Q}_{codebook}}{\arg\min} \ (\|h_j - q\|_2). \tag{1}$$

For training the VQ codebook $\mathbf{Q}_{codebook}$, we use the EMA training and bottleneck layer strategy presented in [18].

Similar to VQMIVC [8], we also explicitly extract the pitch (log-normalized F0) sequence of source speech, denoted as $p \in \mathbb{R}^{1\times T}$, using the World vocoder [19].

The speaker encoder extracts a speaker embedding $\mathbf{s} \in \mathbb{R}^h$, which will be introduced in more details later.

Given $\mathbf{q}$, p and $\mathbf{s}$, a decoder is used to reconstruct the mel-spectrogram of source speech, denoted as $\mathbf{m} \in \mathbb{R}^{k\times T}$, where k is the number of Mel-frequency bins. We first duplicate $\mathbf{q}$ for 4 times and $\mathbf{s}$ for T times, and then frame-wisely concatenate $\mathbf{q}$, p and $\mathbf{s}$ as the input of decoder. The reconstructed mel-spectrogram is denoted as $\hat{\mathbf{m}} \in \mathbb{R}^{k\times T}$. The reconstruction loss is set as $L_{\text{rec}} = \|\hat{\mathbf{m}} - \mathbf{m}\|_2 + \lambda_{\text{VQ}}\|\mathbf{h} - \mathbf{q}\|_2$, where the second item is a VQ loss to minimize the distance between the discrete representation $\mathbf{q}$ and continuous representation $\mathbf{h}$, and λ_{VQ} is a weight to control this loss.

2.1.2. Speaker encoder with contrastive loss

Our speaker encoder is set as the pre-trained model provided by the self-supervised loss-gated-learning (LGL) method [15]. Following only the stage 1 pre-training presented in [15], we train the model from scratch on VCTK, which is then frozen. On top of this model, we add a trainable linear projector to get the speaker embedding $\mathbf{s}$.

In order to impose the speaker similarity between the reconstructed and source utterances, we apply a contrastive loss. For a mini-batch of N utterances, the speaker embedding and reconstructed mel-spectrogram are denoted as $\mathbf{s}_i$ and $\hat{\mathbf{m}}_i, i \in [1, N]$, respectively. Feed $\hat{\mathbf{m}}_i$ into the speaker encoder, we get the speaker embedding $\hat{\mathbf{s}}_i$, which should be close to $\mathbf{s}_i$. On the contrary, the speaker embedding of other utterances in the mini-batch are considered as negative samples of $\hat{\mathbf{s}}_i$. The contrastive loss is defined following [20]

$$L_{\text{spk}} = \frac{1}{N}\sum_{i=1}^{N} -\log \frac{\exp\left(\cos\left(\hat{\mathbf{s}}_i, \mathbf{s}_i\right)\right)}{\sum_{j=1}^{N} \mathbf{1}_{j\neq i} \exp\left(\cos\left(\hat{\mathbf{s}}_i, \mathbf{s}_j\right)\right)} \tag{2}$$

where $\cos(\cdot,\cdot)$ denotes the cosine similarity (dot product) of two vectors, $\mathbf{1}_{k\neq i} \in \{0, 1\}$ is an indicator function.

2.1.3. Discriminator

It is shown in [21] that TTS (and maybe also VC) may suffer from the over-smoothing problem, which could be alleviated by using an additional discriminator. We leverage the discriminator structure and hinge loss from StyleGAN [22].

Moreover, the adaptive loss weight schedule from VQGAN [14] is also used, without which the training may collapse.

The discriminator is denoted as $D(\cdot)$, and GAN loss is

$$L_{\text{GAN}} = \max(0, 1 - D(\mathbf{m})) + \lambda \max(0, 1 + D(\hat{\mathbf{m}})) \quad (3)$$

We remind that $\mathbf{m}$ and $\hat{\mathbf{m}}$ are the real and reconstructed mel-spectrograms, respectively. As presented in [14], the loss weight is set as $\lambda = \nabla[L_{\text{rec}}]/(\nabla[L_{\text{GAN}}] + \delta)$, where $\nabla[\cdot]$ denotes the gradient of its argument w.r.t. the last layer of the decoder, and δ is set for numerical stability. The regular GAN training scheme is used, namely the generator and discriminator are optimized respectively per step.

2.1.4. Perceptual loss

Perceptual loss is widely used in many image generation tasks [14]. We borrow this idea, and adopt a perceptual loss w.r.t. the wav2vec2 representation, as wav2vec2 provides a strong linguistic representation. Specifically, given the reconstructed mel-spectrogram $\hat{\mathbf{m}}$, we feed it to a perceptual network that consists of a trainable ResNet block followed by the frozen wav2vec2 transformer blocks, and obtain a hidden representation $\hat{\mathbf{f}}$. The ℓ_1-norm perceptual loss is defined as $L_{\text{per}} = \|\hat{\mathbf{f}} - \mathbf{f}\|_1$. We remind that $\mathbf{f}$ is the wav2vec2 hidden states of source speech. Experiments show that this perceptual loss results in a great reduction of the objective speech recognition error of the reconstructed mel-spectrogram, which means better linguistic information are preserved.

2.2. Voice Conversion Training

With only the speech reconstruction training, there exists a mismatch between training and inference since speaker information and linguistic information are always from the same speaker during training. To fill this gap, we add a VC branch in the training process. Specifically, in a mini-batch, for one utterance $i \in [1, N]$, its speaker embedding is replaced with the one of one utterance other than i, say i', and the decoder then gives the converted mel-spectrogram $\hat{\mathbf{m}}'_i$. i' is randomly selected by shuffling the original utterance index. Feed $\hat{\mathbf{m}}'_i$ to the speaker encoder described in Section 2.1.2, we obtain the speaker embedding $\hat{\mathbf{s}}'_i$, which should be close to the speaker embedding of utterance i', i.e. $\mathbf{s}_{i'}$, and far away from the one of other utterances. This contrastive loss L_{vcspk} is defined as the same as Eq. (2), except that $\hat{\mathbf{s}}_i$ and $\mathbf{s}_i$ are replaced with $\hat{\mathbf{s}}'_i$ and $\mathbf{s}_{i'}$, respectively. Meanwhile, the perceptual loss is also used, as the converted mel-spectrogram $\hat{\mathbf{m}}'$ should keep the linguistic information of source speech. $\hat{\mathbf{m}}'$ is feed to the same perceptual network as described in Section 2.1.4, and obtain a hidden representation $\hat{\mathbf{f}}'$. The perceptual loss is defined as $L_{\text{per}} = \|\hat{\mathbf{f}}' - \mathbf{f}\|_1$. GAN loss is not applied just for the simplicity of training. Experiments show that this VC branch considerably decreases the objective speech and speaker recognition errors for unseen data. This possibly indicates that the training-inference mismatch is especially problematic for unseen data.

To summary, both the speech reconstruction and VC tasks are performed for one mini-batch, and thus the batch size will be doubled. The overall training loss is then: $L = L_{\text{rec}} + \lambda_{\text{GAN}} L_{\text{GAN}} + \lambda_{\text{spk}} L_{\text{spk}} + L_{\text{per}} + \lambda_{\text{spk}} L_{\text{vcspk}} + L_{\text{vcper}}$, where λs are the respective weights. To accelerate the training, VC is actually performed for only a half of mini-batches, which only leads to a slight performance degradation.

3. EXPERIMENTS

3.1. Experimental setup

Dataset We conduct experiments on the VCTK corpus [23] which has 110 English speakers and around 46 hours of audio. We split them into 100 and 10 speakers respectively for training and development. VCC2020 challenge dataset [24] is used for test, which has 14 speakers and around 2 hours of audio. For the seen-speaker scenario, we randomly select 1000 pairs of samples from VCTK. For the unseen-speaker scenario, we randomly select 100 pairs of samples from VCC2020.

Feature Extraction All utterances are resampled to 16000 Hz. 80-dimensional log mel-spectrograms are extracted with window length (and Fourier transform points) of 400 and hopping size of 160. Mel-spectrograms are normalized to have zero mean and standard deviation.

Optimization The Adam optimizer is adopted to train our model for 400 epochs. The learning rate is first linearly warmed-up from zero to a peak of 7×10^{-5} at the 20-th epoch and then linearly decayed to zero for the rest of 380 epochs. We set λ_{GAN} to 0.75 and λ_{spk} to 0.05. Gradient clip is applied for the discriminator with a norm of 2.

Network Architecture The proposed model includes the following networks: 1) Content encoder consists of 3 convolutional blocks, and each block has 4 ResNet blocks followed by a downsample convolutional layer with a stride of 2 except the last block. The dimension of three blocks are 128, 256, 512, respectively. The trainable VQ codebook includes 192 256-dim discrete codewords. 2) Speaker encoder is the same as the TDNN network presented in [15]. The additional projector is a linear layer projecting the embedding from 192 dim to 256 dim. 3) Decoder includes 16 ResNet blocks, followed by a postnet with 5 convolutional layers. 4) Our discriminator consists of 4 blocks of styleGAN discriminator. 5) As already mentioned, the perceptual network is a ResNet block followed by the frozen wav2vec2 transformer blocks. Finally, we use HIFIGAN [25] as the vocoder to reconstruct waveform from mel-spectrogram. Due to the limit of space, we cannot present all the network details, we will release the code and some audio examples for the proposed model at [1].

[1] https://andyli2022.github.io/dvqvc2022

Table 1. Objective evaluation results.

Method	Seen		Unssen	
	WER%	EER%	WER%	EER%
S2VC [10]	27.1	9.6	20.6	19.2
VQMIVC [8]	32.1	13.9	36.5	45.8
s3prl-VC [11]	**8.4**	30.8	**6.2**	38.5
DVQVC (ours)	10.2	**9.2**	7.4	**12.9**

Evaluation metric We compare with three advanced VC models, i.e. VQMIVC [8], s3prl-VC [11] and S2VC [10]. All of them are trained and tested using the same datasets as the proposed model. For objective evaluation, we conduct ASR using the wav2vec2's offcial tool to evaluate the preserved linguistic information, using the metric of WER (word error rate). The advanced speaker verification tool wespeaker [2] is used to evaluate the speaker similarity using the metric of EER (equal error rate). For subjective evaluations, we conduct mean opinion score (MOS) test using MUSHRA [26], including naturalness MOS (nMOS) and speaker similarity MOS (sMOS). For each test, we randomly select 20 samples.

3.2. VC performance

Table 1 shows the objective evaluation results. The proposed model achieves the lowest EER and close WER with s3prl-VC. This means the proposed model is able to adequately convert the source speech to the target speaker, and meanwhile to maintain a high speech intelligibility. Table 2 shows the subjective evaluation results. The proposed method achieves comparable nMOS scores with s3prl-VC, and much better sMOS scores than the other methods. Again, this verifies that the proposed model can adequately perform voice conversion, and meanwhile maintain a high speech quality. In addition, both the objective and subjective measures show the good generalization capability of the proposed model to unseen speakers. Overall, the good performance indicates that the proposed model can effectively disentangle speaker and linguistic information, and meanwhile speaker and linguistic information are both well preserved to not distort the converted speech quality.

3.3. Ablation studies

Table 3 shows the results of ablation studies. We first verify the effectiveness of the training strategies by removing each of them. As for the '- all above four' method (only the reconstruction loss is used), WER is high, and the EER of unseen scenario is also high. The perceptual loss is the most important strategy for reducing the WER, by pushing the reconstructed/converted speech to have good linguistic information. Discriminator improves all metrics, which means the discrimination between real and generated speech is also useful for VC. The speaker loss and VC branch mainly improve the performance of unseen data, and equivalently the generalization capability. On the base of '- all above four', we further testify the effectiveness of sub-networks. When wav2vec2 is removed (and directly use mel-spectrogram as input), the huge performance degradation (especially for WER) indicates that the content encoder alone is difficult to extract high-quality linguistic feature. When VQ is removed, the WERs get better while the EERs get worse. VQ squeezes out speaker information from the content encoder, and leads to less speaker leakage but also a loss of linguistic information. When the LGL speaker encoder is also removed (use the same speaker encoder as s3prl-VC), the EERs further increase, which means a better speaker embedding is helpful for improving the quality of converted speaker.

Overall, the strategies used in the proposed model are all playing positive roles. Their contributions are briefly summarized as follows, which are the key for the success of the proposed model. wav2vec2 provides good linguistic representation, based on which VQ and LGL speaker encoder further disentangle the linguistic and speaker representations. The perceptual loss compensates the loss of linguistic information. The speaker loss and VC branch improve the rationality of the task setting, which is helpful for speaker generalization. The discriminator makes the generated mel-spectrograms more real in every aspects.

Table 2. Subjective evaluation results.

Method	Seen		Unssen	
	nMOS	sMOS	nMOS	sMOS
S2VC [10]	2.95 ± 0.92	3.26 ± 0.98	2.94 ± 0.96	3.20 ± 1.00
VQMIVC [8]	3.11 ± 0.95	3.19 ± 0.95	3.125 ± 1.04	2.84 ± 0.93
s3prl-VC [11]	$\mathbf{4.01} \pm 0.81$	3.88 ± 0.86	4.21 ± 0.74	3.97 ± 0.76
DVQVC (ours)	3.98 ± 0.85	$\mathbf{4.25} \pm 0.78$	$\mathbf{4.24} \pm 0.70$	$\mathbf{4.32} \pm 0.63$

Table 3. Ablation studies.

Method	Seen		Unssen	
	WER%	EER%	WER%	EER%
DVQVC	10.2	9.2	7.4	12.9
- perceptual loss	22.9	6.9	18.2	15.1
- speaker loss	10.2	8.2	9.2	17.6
- discriminator	11.3	10.2	8.6	16.4
- VC branch	10.5	7.8	9.8	16.5
- all above four	19.1	9.2	17.7	15.3
- wav2vec2	82.1	12.2	73.6	21.2
- VQ	8.7	11.8	5.6	24.7
- LGL spk-enc	7.8	12.8	5.6	31.2

4. CONCLUSIONS

This paper proposes a new zero-shot voice conversion system, named DVQVC. By properly integrating the SSL speech feature, VQ and LGL speaker encoder, and designing a perceptual loss, GAN loss and speaker contrastive loss, DVQVC finally conducts excellent speaker conversion and meanwhile preserves outstanding speech quality and intelligibility.

[2] https://github.com/wenet-e2e/wespeaker

5. REFERENCES

[1] N. Tomashenko, B. M. L. Srivastava, X. Wang, E. Vincent, A. Nautsch, *et al.*, "Introducing the VoicePrivacy Initiative," in *Proc. Interspeech 2020*, pp. 1693–1697, 2020.

[2] D. Wang, S. Liu, L. Sun, X. Wu, X. Liu, *et al.*, "Learning Explicit Prosody Models and Deep Speaker Embeddings for Atypical Voice Conversion," in *Proc. Interspeech 2021*, pp. 4813–4817, 2021.

[3] A. Gabry's, G. Huybrechts, M. S. Ribeiro, C. M. Chien, J. Roth, *et al.*, "Voice filter: Few-shot text-to-speech speaker adaptation using voice conversion as a post-processing module," *ICASSP 2022*, pp. 7902–7906, 2022.

[4] S. Liu, Y. Cao, S. Kang, N. Hu, X. Liu, *et al.*, "Transferring Source Style in Non-Parallel Voice Conversion," in *Proc. Interspeech 2020*, pp. 4721–4725, 2020.

[5] Y. Jia, Y. Zhang, R. Weiss, Q. Wang, J. Shen, F. Ren, *et al.*, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," *Advances in neural information processing systems*, vol. 31, 2018.

[6] S. Zhao, H. Wang, T. H. Nguyen, and B. Ma, "Towards natural and controllable cross-lingual voice conversion based on neural tts model and phonetic posteriorgram," *ICASSP 2021*, pp. 5969–5973, 2021.

[7] D.-Y. Wu, Y.-H. Chen, and H.-Y. Lee, "Vqvc+: One-shot voice conversion by vector quantization and u-net architecture," *arXiv preprint arXiv:2006.04154*, 2020.

[8] D. Wang, L. Deng, Y. T. Yeung, X. Chen, X. Liu, and H. Meng, "Vqmivc: Vector quantization and mutual information-based unsupervised speech representation disentanglement for one-shot voice conversion," *arXiv preprint arXiv:2106.10132*, 2021.

[9] H. Tang, X. Zhang, J. Wang, N. Cheng, and J. Xiao, "Avqvc: One-shot voice conversion by vector quantization with applying contrastive learning," *ICASSP 2022*, pp. 4613–4617, 2022.

[10] J.-h. Lin, Y. Y. Lin, C.-M. Chien, and H.-y. Lee, "S2vc: a framework for any-to-any voice conversion with self-supervised pretrained representations," *arXiv preprint arXiv:2104.02901*, 2021.

[11] W.-C. Huang, S. wen Yang, T. Hayashi, H. yi Lee, S. Watanabe, *et al.*, "S3prl-vc: Open-source voice conversion framework with self-supervised speech representations," *ICASSP 2022*, pp. 6552–6556, 2022.

[12] A. Baevski, S. Schneider, and M. Auli, "vq-wav2vec: Self-supervised learning of discrete speech representations," *arXiv preprint arXiv:1910.05453*, 2019.

[13] S. Chen, Y. Wu, C. Wang, S. Liu, Z. Chen, *et al.*, "Why does self-supervised learning for speech recognition benefit speaker recognition?," *arXiv preprint arXiv:2204.12765*, 2022.

[14] D.-H. Im and Y. seok Seo, "Generating face images using vqgan and sparse transformer," *2021 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 1642–1644, 2021.

[15] R. Tao, K.-A. Lee, R. K. Das, V. Hautamaki, and H. Li, "Self-supervised speaker recognition with loss-gated learning," *ICASSP 2022*, pp. 6142–6146, 2022.

[16] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in Neural Information Processing Systems*, vol. 33, pp. 12449–12460, 2020.

[17] A. H. Liu, W.-N. Hsu, M. Auli, and A. Baevski, "Towards end-to-end unsupervised speech recognition," *arXiv preprint arXiv:2204.02492*, 2022.

[18] J. Yu, X. Li, J. Y. Koh, H. Zhang, R. Pang, *et al.*, "Vector-quantized image modeling with improved vqgan," *arXiv preprint arXiv:2110.04627*, 2021.

[19] M. Morise, F. Yokomori, and K. Ozawa, "World: a vocoder-based high-quality speech synthesis system for real-time applications," *IEICE TRANSACTIONS on Information and Systems*, vol. 99, no. 7, pp. 1877–1884, 2016.

[20] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*, pp. 1597–1607, PMLR, 2020.

[21] Y. Ren, X. Tan, T. Qin, Z. Zhao, and T.-Y. Liu, "Revisiting over-smoothness in text to speech," *arXiv preprint arXiv:2202.13066*, 2022.

[22] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, *et al.*, "Analyzing and improving the image quality of stylegan," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8107–8116, 2020.

[23] 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.

[24] Y. Zhao, W.-C. Huang, X. Tian, J. Yamagishi, R. K. Das, T. Kinnunen, Z. Ling, and T. Toda, "Voice conversion challenge 2020: Intra-lingual semi-parallel and cross-lingual voice conversion," *arXiv preprint arXiv:2008.12527*, 2020.

[25] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17022–17033, 2020.

[26] M. Schoeffler, S. Bartoschek, F.-R. Stöter, M. Roess, S. Westphal, *et al.*, "webmushra — a comprehensive framework for web-based listening tests," *Journal of open research software*, vol. 6, 2018.