AVQVC: ONE-SHOT VOICE CONVERSION BY VECTOR QUANTIZATION WITH APPLYING CONTRASTIVE LEARNING

Huaizhen Tang[1,2]*, Xulong Zhang*[1]*, Jianzong Wang*[1*]*, Ning Cheng*[1]*, Jing Xiao*[1]

[1]Ping An Technology (Shenzhen) Co., Ltd.
[2]University of Science and Technology of China

ABSTRACT

Voice Conversion(VC) refers to changing the timbre of a speech while retaining the discourse content. Recently, many works have focused on disentangle-based learning techniques to separate the timbre and the linguistic content information from a speech signal. Once successful, voice conversion will be feasible and straightforward. This paper proposed a novel one-shot voice conversion framework based on vector quantization voice conversion (VQVC) and AutoVC, called **AVQVC**. A new training method is applied to VQVC to separate content and timbre information from speech more effectively. The result shows that this approach has better performance than VQVC in separating content and timbre to improve the sound quality of generated speech.

Index Terms— speech synthesis, contrastive learning, voice conversion, vector quantization

1. INTRODUCTION

Voice conversion (VC) aims to convert an utterance of a source speaker to another utterance of a target person by keeping the content in the original utterance and replacing it with the vocal features from the target speaker. Recently, considerable effort was spent on the topic of VC. Up to now, many methods have been applied in VC successfully[1, 2, 3, 4]. Among them, some approaches of VC require training a model for each paired speaker using parallel corpora, which limits the ability to produce natural speech for a target speaker without enough pair source-target data. To address this problem, more and more attention has focused on non-parallel VC systems.

Recently, many Generative Adversarial Networks (GAN) [5, 6, 7, 8, 9] have been successfully applied to non-parallel VC. These GAN-based models jointly train a generator network with a discriminator. Where adversarial loss derived from the discriminator encourages the generator outputs to build indistinguishable from real speech. Due to the cycle consistency training, GAN-based VC models can be trained with non-parallel speech datasets.

Besides, learning discrete representations of speech has also gathered much attention. Vector Quantization (VQ), an effective data compression technology, can quantify continuous data into discrete data. Previous studies have confirmed that the quantized discrete data from the input continuous speech data is closely related to the phoneme information[10]. Recently, VQVC [11] has been proposed to learn to disentangle the content and speaker information with reconstruction loss only. Then, VQVC+ [12] was also proposed to improve the conversion performance of VQVC by adding the U-Net architecture within an auto-encoder-based VC system.

There is also another line of research focus on learning continuous speech representations via predicting context information[13, 14]. VQ-wav2vec[15] combines this line of research with VQ to learn discrete speech representations, and Koshizuka T *et al.* [16] introduced pre-trained VQ-wav2vec to achieve any-to-many voice conversion. Meanwhile, some research also tries to combine VQ with other existing work. For example, VQ-VAE [17] combines VQ and VAE to improve the learning ability to disentangle the content and timbre information. VQ-CPC [18] was also proposed by combining VQ and CPC.

Unfortunately, all these VQ-based models have their inherent disadvantages. For example, the training of VQVC is simple and fast enough, but the audio quantity of this method is very poor. VQVC+ improves the conversion performance of VQVC while sacrificing the simple network structure. VQ-VAE, VQ-CPC, and VQ-wav2vec have the same problem. That is, they all introduce other network structures, which makes the model very complex.

Recently, the emerging of the model called AutoVC has given us great inspiration. By applying a speaker encoder pretrained with GE2E loss [19, 20, 21], maximizing the embedding similarity between different utterances of the same speaker, and minimizing the similarity between different speakers, AutoVC can easily get the ideal speaker embedding. To take advantage of this work, we can disentangle the content and speaker embedding more reasonably. In this way, we can improve the conversion performance of VQVC without increasing any algorithm complexity of the model.

This paper proposed a novel voice conversion framework that combines the AutoVC and VQVC systems, named

[*] Corresponding author: Jianzong Wang, (jzwang@188.com).

AVQVC. Specifically, we redesign the training of VQVC by learning from the idea of AutoVC to force our model to separate the linguistic and timbre information correctly. Experiment results are carried out on the VCTK datasets. Our main contributions are as follows:

- We applied a new training method in VQVC to guide discrete vector to be closer to the content features while the mean difference between encoder output and discrete vector is encouraged to be more closer to the speaker information;
- Compared to AutoVC, we can quickly get content embedding and speaker embedding with only one codebook structure so that we do not need to introduce a pre-trained speaker encoder like AutoVC;

2. METHOD

2.1. VQVC

VQVC designed a simple framework to disentangle the content embedding and speaker embedding with only one reconstruction loss. As shown in Figure 1, the framework of VQVC contains three network structures: an encoder to extract latent features from the input speech, a learnable codebook that quantizes continuous data into discrete data, and a decoder that produces the converted speech from the content and speaker embedding. We regard the discrete data generated by Codebook as content embedding. Then, we can quickly get speaker embedding from the mean difference between encoder output and discrete data. The discrete content embedding $\boldsymbol{C_x}$ and the speaker embedding $\boldsymbol{S}_x$ can be derived as

$$C_x = VQ(enc(x)) \qquad S_x = E_t[enc(x) - C_x] \tag{1}$$

Where we define $enc(\cdot)$ as the encoder, The expectation E_t takes on the segment length on latent space, and VQ was denoted as the quantization function which can quantize sequence of continuous data into the closest discrete code. Specifically, if we define V as a sequence of continuous data, that is, $\boldsymbol{V} = v_1, v_2, ...v_T$. Then $VQ(\boldsymbol{V})$ can be described as

$$VQ(\boldsymbol{V}) = q_1, q_2, ...q_T \tag{2}$$

$$q_j = \arg \min_{q \in Codebook} (\|v_j - q\|_2^2) \tag{3}$$

In the training phase, one utterance x_i was selected randomly to do the reconstruction task. The latent-code loss function was optimized to minimize the distance between the discrete code and the continuous embedding. Besides, the self-reconstruction loss function was designed to constrain the model to find a proper balance between removing speaker information and retaining linguistic content. They can be expressed as:

$$\mathcal{L}_{\text{latent}} = \mathbb{E}[\|enc(x) - C_x\|_2^2] \tag{4}$$

$$\mathcal{L}_{\text{recon}} = \mathbb{E}[\|\hat{x}_{i \to i} - x_i\|_1^1] \tag{5}$$

$$\hat{x}_{i \to i} = Decoder(c_{x_i} + s_{x_i}) \tag{6}$$

In conversion, two utterances from different speakers were chosen to be the source and target speech, respectively. We fed these two utterances into the trained model to get content embedding and speaker embedding, respectively. Then, the content embedding of the source speaker and the target speaker's speaker embedding were put into the decoder together to get the conversion audio from the decoder output.

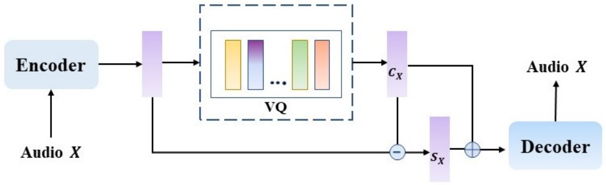

Fig. 1. Framework of VQVC. $\boldsymbol{C}_x$ is the discrete code which is produced by $codebook$. $\boldsymbol{S}_x$ denotes speaker embedding, and it is gengrated from the mean difference between encoder output and $\boldsymbol{C}_x$.

2.2. AVQVC

In this section, we will introduce the core idea of our method. As illustrated in Figure 2, the composition network structure of our model is similar to VQVC, but the training method is completely different. Specifically, the input of our model contains three sentences instead of one. And they are represented by x_1, x_2, and x_3 respectively. Among them, x_1 and x_2 are produced by the same speaker, but their text content is different, and x_3 is generated by another speaker. We assume that our VQ model can separate content embedding and speaker embedding correctly, which means that when x_1 and x_2 are sent to our VQ model after the same processing, their speaker embedding should be the same. Furthermore, when we put x_1 and x_3 (or x_2 and x_3) into our VQ model after the same processing, their speaker embedding should be far different. Based on this assumption, we designed a new training program.

In training phase, we input x_1, x_2, and x_3 into the same model at the same time, then we can easily get $\boldsymbol{C}_{x_i}$ and $\boldsymbol{S}_{x_i}$ $(i = 1, 2, 3)$, Then we exchange $\boldsymbol{S}_{x_1}$ and $\boldsymbol{S}_{x_2}$ to do a novel reconstruction task. Since $\boldsymbol{S}_{x_1}$ and $\boldsymbol{S}_{x_2}$ are expected to be the same, the new reconstruction speech x_1', x_2' should be as close to x_1, x_2 as possible. Here we still use self-reconstruction loss and latent-code loss to constrain VQ model. They can be expressed as

$$\mathcal{L}_{\text{recon}} = \|x_1' - x_1\|_1^1 + \|x_2' - x_2\|_1^1 + \|x_3' - x_3\|_1^1 \tag{7}$$

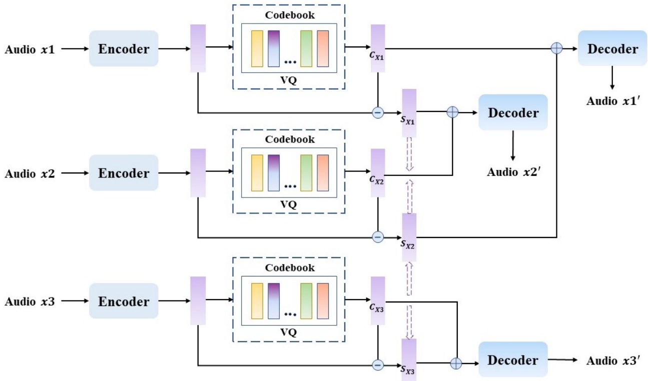

Fig. 2. Framework of AVQVC. Both x_1 and x_2 are produced by the same speaker, but their text content are different, while x_3 belongs to another speaker. $\boldsymbol{C}_X$ is a discrete variable generated by looking up the $codebook$. And, $\boldsymbol{S}_X$ denotes speaker embedding, which is produced by the mean difference between encoder output and $\boldsymbol{C}_X$.

$$\mathcal{L}_{\text{latent}} = \|enc(x_1) - C_{x_1}\|_2^2 \\ + \|enc(x_2) - C_{x_2}\|_2^2 + \|enc(x_3) - C_{x_3}\|_2^2 \quad (8)$$

Where x_1' produced by $\boldsymbol{C}_{x_1}$ and $\boldsymbol{S}_{x_2}$, x_2' generated from $\boldsymbol{C}_{x_2}$ and $\boldsymbol{S}_{x_1}$. And, x_3' is special, it comes from $\boldsymbol{C}_{x_3}$ and $\boldsymbol{S}_{x_3}$, both of which are all produced by x_3 itself. That is

$$x_3' = Decoder(C_{x_3} + S_{x_3}). \quad (9)$$

In addition, we also design speaker-loss function and diff-speaker-loss function to encourage discrete data to be as close to content embedding as possible. Then, the mean difference between continuous and discrete data will naturally become closer and closer to speaker embedding. Specifically, for $\boldsymbol{S}_{X_1}$ and $\boldsymbol{S}_{X_2}$, since their speaker is the same one, we expect their speaker embeddings to be as close as possible. Similarly, because $\boldsymbol{S}_{X_1}$ and $\boldsymbol{S}_{X_3}$ (or $\boldsymbol{S}_{X_2}$ and $\boldsymbol{S}_{X_3}$) are produced by different speakers, we expected their speaker embeddings to be as different as possible. These two loss functions can be computed as

$$\mathcal{L}_{\text{speaker}} = \|S_{x_2} - S_{x_1}\|_1^1 \quad (10)$$

$$\mathcal{L}_{\text{diff}} = -(\|S_{x_2} - S_{x_3}\|_1^1 + \|S_{x_1} - S_{x_3}\|_1^1) \quad (11)$$

Then the full objective function can be formulated as

$$L = \mathcal{L}_{\text{recon}} + \alpha\mathcal{L}_{\text{latent}} + \beta\mathcal{L}_{\text{speaker}} + \lambda\mathcal{L}_{\text{diff}} \quad (12)$$

The inference phase of AVQVC is completely consistent with that of VQVC. One utterance of the source speaker and the target speaker is selected to get content embedding and speaker embedding respectively. After that, we input them into the decoder, and the conversion speech is then generated.

3. EXPERIMENTS

3.1. Experimental Setup

We evaluated our proposed method on the VCTK Corpus[22], which contains 46 hours of speech data produced by 109 English speakers from different countries. In our work, the entire dataset is randomly divided into three sets: 90 speaker recordings for training, 10 speaker recordings for evaluation and other 4167 recordings from 9 speakers for testing. And, the sampling rate of all recordings is 16khz, and the mel-spectrograms are computed through a short-time Fourier transform (STFT) with Hann windowing, where 1024 for FFT size, 1024 for window size and 256 for hop size. The STFT magnitude is transformed to the mel scale using 80 channel mel filter bank spanning 90 Hz to 7.6 kHz.

We train AVQVC model with batch size of sixteen for 1M steps on one NVIDIA V100 GPU, and ADAM optimizer was used with $\beta_1 = 0.9, \beta_2 = 0.98$. The weights in Eq.(13) are set to $\alpha = 0.02, \beta = 0.03, \lambda = 0.02$. It is worth noting that with the increase of λ , the training of our model becomes somewhat unstable, and when λ is too small, it will be difficult for our model to remove the speaker information from the input speech. In fact, in order to ensure the stability of model training, when the value of $\mathcal{L}_{\text{diff}}$ is five times more than $\mathcal{L}_{\text{recon}}$, we will reduce $\lambda = 0.01$. At the same time, we will also increase $\beta = 0.05$ and change the weights of recon-loss to 2, which can encourage the model to learn codebook closer to continuous data so as to improve the converted audio quality.

3.2. Comparison

Here we will evaluate the performance of our method in traditional VC tasks and one-shot VC tasks. Specifically, traditional VC means that both the selected source speaker and the target speaker already exist in the training set. And one-shot VC refers to a new task only needs one utterance from the source speaker and target speaker, and both these two speakers do not need to appear in the training set. To compare the performance of VC between our method and other previous works, the Mel-Cepstral Distortion(MCD) between converted speech and the ground truth target speech as our objective evaluation and two subjective evaluation methods were also introduced. One is the mean opinion score(MOS) test, which is used to evaluate the quality of converted speech. Specifically, we invited 12 humans (seven males and five females) participants to evaluate the quality of some converted speech generated from different models. After hearing each speech, the subjects should choose a score from 1-5 points of the naturalness of the converted speech. The higher the score, the better they think the audio quality of the speech. The other is the voice similarity score(VSS) test, which measures how similar the timbre of the converted voice is to that of the ground truth. And its scoring mechanism is the same as that of MOS. The higher the score, the more similar the tone

Table 1. Comparison of different models in traditional VC and one-shot vc.

Methods	Traditional VC			One-Shot VC			
	MCD	MOS	VSS	MCD	MOS	VSS	MODEL-SIZE
VQVC	8.16 ± 0.31	2.28 ± 0.99	3.47 ± 0.82	8.12 ± 0.14	2.06 ± 0.84	2.97 ± 0.75	**5.71M**
VQVC+	7.08 ± 0.22	3.31 ± 0.90	3.42 ± 0.85	8.41 ± 0.08	2.75 ± 0.84	3.11 ± 0.88	388M
AutoVC	**4.34 ± 0.12**	**3.81 ± 1.14**	3.45 ± 0.76	7.66 ± 0.17	2.61 ± 0.73	2.91 ± 0.72	339M
StarGAN-VC2	6.28 ± 0.09	3.45 ± 1.01	3.59 ± 0.87	—	—	—	56.45M
AVQVC(512)	5.19 ± 0.29	3.57 ± 0.91	**3.70 ± 0.71**	**5.04 ± 0.13**	**3.20 ± 0.91**	**3.29 ± 0.64**	5.77M

between the converted speech and the target speech. AutoVC, VQVC, VQVC+, StarGAN-VC2 were chosen as baselines. And the result shows in table 1.

The result shows that with the new training method, the quality of audio converted by VQVC has been significantly improved. Compared with other VQ-based models, the speech quality generated by our method is equivalent and slightly worse than that of AutoVC, but in VSS Test, our performance is better than AutoVC. Most importantly, compared with VQVC, our method only adds few parameters, which makes our model much simpler and faster than AutoVC, StarGAN-VC2, or other VQ-based models.

In addition, by applying a well-known open-source speech detection toolkit, *Resemblyzer* (`https://github.com/resemble-ai/Resemblyzer`), we conduct a fake speech detection test to compare the quality and similarity of the converted speeches from different models respectively against ground truth reference audio. The higher the score, the better the quality and similarity of the converted speech. The results are shown in Table 2.

Table 2. Comparison of different methods in traditional VC tasks.

Method	Score
AVQVC	**0.72 ± 0.24**
AutoVC	0.70 ± 0.43
VQVC+	0.63 ± 0.31

The result shown in Table 2 shows that the converted speech produced by AVQVC is more similar to the ground truth than that of AutoVC and VQVC+. It indicates AVQVC outperforms AutoVC and VQVC+ in the VC tasks.

Furthermore, we also evaluate the performance of different models in the one-shot VC task. Since StarGAN-VC2 can not achieve the VC task for unseen speakers, AutoVC, VQVC, and VQVC+ were chosen as our baseline models. The results in table 1 show that with only one utterance of unseen speakers, the performance of AutoVC is greatly reduced. Previous studies have reported this phenomenon [23]. While in the same case, our method still has a good performance, indicating that our model has strong adaptability under both conditions.

3.3. Codebook Size

In addition, we also focused on the choice of codebook size. In VQVC, 256 is selected as an appropriate number of discrete codes. And in our work, since we add some new loss functions to constraint the codebook, we don't need to worry that using a large codebook size will include speaker information, so we conduct many comparison experiments to find the proper number of codebook sizes. Finally, 512 is selected. And we will show the MOS and VSS results under different codebook sizes in Figure 3.

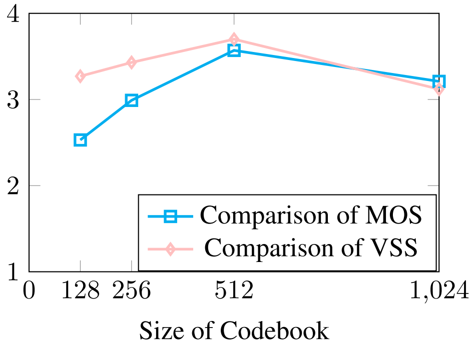

Fig. 3. Performance comparison of MOS and VSS under different Codebook sizes

As illustrated in Figure 3, when the size of the codebook is 512, the value of MOS and VSS are all significantly improved—noted that when codebook size exceeds 512, the performance of our method also decreases. We hypothesize that when the codebook's size is too large, the discrete vector will inevitably contain some timbre information, the loss function we designed will lose its constraint and our model will finally degenerate into the original VQVC.

4. CONCLUSION

In this paper, we proposed a new model to apply a new training method to VQVC to improve the performance of voice conversion. We conducted both objective and subjective evaluations to evaluate the performance of our method in different VC tasks. The result shows that with the new training method applied to VQVC, the quality of converted audio speech and the similarity with the target voice has significantly improved.

5. ACKNOWLEDGEMENT

This paper is supported by the Key Research and Development Program of Guangdong Province No. 2021B0101400003 and the National Key Research and Development Program of China under grant No. 2018YFB0204403.

6. REFERENCES

[1] Yannis Stylianou, Olivier Cappé, and Eric Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Transactions on speech and audio processing*, vol. 6, no. 2, pp. 131–142, 1998.

[2] Riku Arakawa, Shinnnosuke Takamichi, and Hiroshi Saruwatari, "Implementation of dnn-based real-time voice conversion and its improvements by audio data augmentation and mask-shaped device," in *Proc. SSW10*, 2019, pp. 93–98.

[3] Chin-Cheng Hsu, Hsin-Te Hwang, Yi-Chiao Wu, Yu Tsao, and Hsin-Min Wang, "Voice conversion from non-parallel corpora using variational auto-encoder," in *2016 APSIPA*. IEEE, 2016, pp. 1–6.

[4] Wen-Chin Huang, Hsin-Te Hwang, Yu-Huai Peng, Yu Tsao, and Hsin-Min Wang, "Voice conversion based on cross-domain features using variational auto encoders," in *ISCSLP*. IEEE, 2018, pp. 51–55.

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[6] Chin-Cheng Hsu, Hsin-Te Hwang, Yi-Chiao Wu, Yu Tsao, and Hsin-Min Wang, "Voice conversion from unaligned corpora using variational autoencoding wasserstein generative adversarial networks," in *Interspeech 2017*. 2017, pp. 3364–3368, ISCA.

[7] Takuhiro Kaneko and Hirokazu Kameoka, "Cyclegan-vc: Non-parallel voice conversion using cycle-consistent adversarial networks," in *2018 EUSIPCO*. IEEE, 2018, pp. 2100–2104.

[8] Hirokazu Kameoka, Takuhiro Kaneko, Kou Tanaka, and Nobukatsu Hojo, "Stargan-vc: Non-parallel many-to-many voice conversion using star generative adversarial networks," in *SLT*. IEEE, 2018, pp. 266–273.

[9] Takuhiro Kaneko, Hirokazu Kameoka, Kou Tanaka, and Nobukatsu Hojo, "Stargan-vc2: Rethinking conditional methods for stargan-based voice conversion," *Proc. Interspeech 2019*, pp. 679–683, 2019.

[10] Jan Chorowski, Ron J Weiss, Samy Bengio, and Aäron van den Oord, "Unsupervised speech representation learning using wavenet autoencoders," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 27, no. 12, pp. 2041–2053, 2019.

[11] Da-Yi Wu and Hung-yi Lee, "One-shot voice conversion by vector quantization," in *ICASSP 2020-2020*. IEEE, 2020, pp. 7734–7738.

[12] Da-Yi Wu, Yen-Hao Chen, and Hung-yi Lee, "Vqvc+: One-shot voice conversion by vector quantization and u-net architecture," *Proc. Interspeech 2020*, pp. 4691–4695, 2020.

[13] Yu-An Chung and James Glass, "Speech2vec: A sequence-to-sequence framework for learning word embeddings from speech," *Proc. Interspeech 2018*, pp. 811–815, 2018.

[14] Aaron van den Oord, Yazhe Li, and Oriol Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[15] Alexei Baevski, Steffen Schneider, and Michael Auli, "vq-wav2vec: Self-supervised learning of discrete speech representations," in *ICLR*, 2019.

[16] Takeshi Koshizuka, Hidefumi Ohmura, and Kouichi Katsurada, "A vocoder-free any-to-many voice conversion using pre-trained vq-wav2vec," *IEICE Technical Report; IEICE Tech. Rep.*, vol. 120, no. 399, pp. 176–181, 2021.

[17] Shaojin Ding and Ricardo Gutierrez-Osuna, "Group latent embedding for vector quantized variational autoencoder in non-parallel voice conversion.," in *INTERSPEECH*, 2019, pp. 724–728.

[18] Benjamin van Niekerk, Leanne Nortje, and Herman Kamper, "Vector-quantized neural networks for acoustic unit discovery in the zerospeech 2020 challenge," in *Interspeech 2020*. 2020, pp. 4836–4840, ISCA.

[19] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno, "Generalized end-to-end loss for speaker verification," in *2018 ICASSP*. IEEE, 2018, pp. 4879–4883.

[20] Ye Jia, Yu Zhang, Ron Weiss, Quan Wang, Jonathan Shen, Fei Ren, Patrick Nguyen, Ruoming Pang, Ignacio Lopez Moreno, Yonghui Wu, et al., "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," *Advances in neural information processing systems*, vol. 31, 2018.

[21] Huaizhen Tang, Xulong Zhang, Jianzong Wang, Ning Cheng, Zhen Zeng, Edward Xiao, and Jing Xiao, "Tgavc: Improving autoencoder voice conversion with text-guided and adversarial training," in *ASRU*. IEEE, 2021, pp. 938–945.

[22] Christophe Veaux, Junichi Yamagishi, Kirsten MacDonald, et al., "Superseded-cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit," 2016.

[23] Zhiyuan Tan, Jianguo Wei, Junhai Xu, Yuqing He, and Wenhuan Lu, "Zero-shot voice conversion with adjusted speaker embeddings and simple acoustic features," in *ICASSP 2021*. 2021, pp. 5964–5968, IEEE.