INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

TVQVC: Transformer based Vector Quantized Variational Autoencoder with CTC loss for Voice Conversion

Ziyi Chen[1,2], *Pengyuan Zhang*[1,2,*]

[1]Key Laboratory of Speech Acoustics & Content Understanding, Institute of Acoustics, CAS, China
[2]University of Chinese Academy of Sciences, Beijing, China

{chenziyi,zhangpengyuan}@hccl.ioa.ac.cn

Abstract

Techniques of voice conversion(VC) aim to modify the speaker identity and style of an utterance while preserving the linguistic content. Although there are lots of VC methods, the state of the art of VC is still cascading automatic speech recognition(ASR) and text-to-speech(TTS). This paper presents a new structure of vector-quantized autoencoder based on transformer with CTC loss for non-parallel VC, which inspired by cascading ASR and TTS VC method. Our proposed method combines CTC loss and vector quantization to get high-level linguistic information without speaker information. Objective and subjective evaluations on the mandarin datasets show that the converted speech of our proposed model is better than baselines on naturalness, rhythm and speaker similarity.

Index Terms: voice conversion, vector quantization, transformer, ctc

1. Introduction

In general, the speech of human conveys lots of information, such as linguistic information and speaker information. The task of voice conversion is to convert the timbre and rhythm of the source speaker's utterance to target speaker's while preserving the linguistic information. Voice conversion can be broadly grouped into parallel and non-parallel methods, which are distinguished by their training data.

Traditionally, voice conversion focuses on using parallel training data to take one-to-one or many-to-one transformation. It requires pair of speech samples with the same linguistic content. And it usually focuses on mapping source spectral features to target, such as gaussian mixture model(GMM)[1, 2], frequency wrapping[3, 4] and some DNN based models[5, 6, 7].

However, collecting parallel data is challenging. So in recent years, there are plenty of researches on non-parallel voice conversion. The intuition of non-parallel voice conversion is how to disentangle the context and speaker information in high-dimension space. There are several methods for non-parallel voice conversion. Flow based model such as blow[8], which is directly applied on time domain not frequency domain. Generative adversarial network[9] and its variants, for example CycleGAN-VC[10, 11] and StarGAN-VC[12, 13], which use generator or conditional generator to transform the source speaker's features to target speaker's features directly. There are also some autoencoder based methods, such as AutoVC[14, 15], VQVC[16, 17] and VAEVC[18], which use autoencoder to create information bottleneck between encoder and decoder to remove speaker information. And directly using ASR to remove speaker information is intuitive, so there are several PPG[6, 19] or ASR[20, 21] based VC models, which take advantage of pre-trained ASR model to generate speaker independent information.

This work was inspired by cascading ASR and TTS voice conversion. There are lots of ASR methods based on Transformer[22] and song[23] utilized conformer based ASR model to extract PPGs for converting source speech to linguistic information without speaker information. But we can't guarantee there is no residual speaker dependent information in PPGs, which may influence the quality of converted speech. The core of non-parallel voice conversion task is how to disentangle speaker representations and content representations respectively. Inspired by the deep vector quantization model, which has been observed that the discrete latent codes learned from vector-quantized autoencoder are highly related to the phonemes[24, 25]. So we proposed a novel structure of transformer based vector-quantized autoencoder with CTC loss and reconstruction loss. The encoder helps remove most of speaker dependent information to get context information. The vector-quantized layer can be seen as deep clustering to covert contiguous embeddings into discrete embeddings for removing residual speaker information. Our contribution is three-fold:

- We present a new method TVQVC to disentangle the content information and speaker information.
- We combine some modules of the state of the art models in ASR and TTS to achieve a lightweight model for voice conversion.
- We use CTC loss to help accelerate model to converge and use vector quantization to remove residual speaker information.

2. Methods

In this section, we describe the model architecture and training procedure of our proposed method. Figure 1 and Figure 2 show the overall pipeline of baseline model and our proposed model respectively. Our model structure was inspired by PPG based VC, which utilizes the encoder of hybrid CTC/attention transformer based model to extract PPGs and converts PPGs to mel-spectrum by TTS acoustic model. We will elaborate the structure in detail below.

2.1. VQVAE

We adopted the VQVAE model[1] as the baseline model[24], which is shown in Figure 1. The baseline was based on the wavenet[26] autoencoder proposed in [25]. Here a lightweight RNN-based decoder replaces the wavenet decoder. The core idea is that content information can be represented by discrete

* corresponding author

[1]https://github.com/bshall/ZeroSpeech

http://dx.doi.org/10.21437/Interspeech.2021-1301

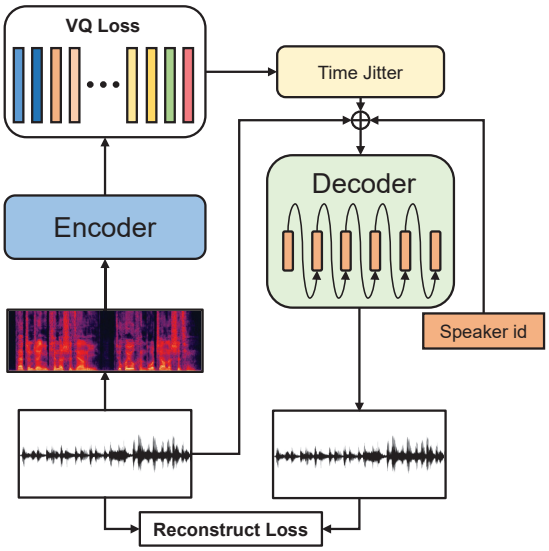

Figure 1: *The structure of VQVAE. The encoder of VQVAE is a stack of several convolution blocks and the decoder of VQVAE is a lightweight RNN-based model.*

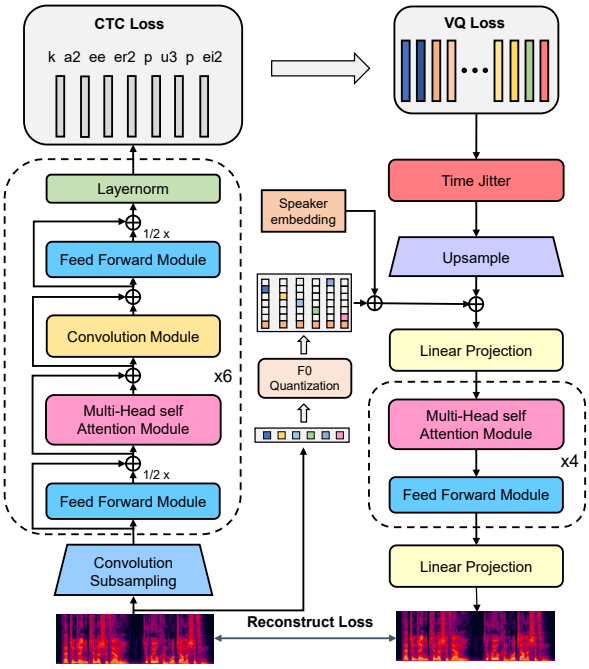

Figure 2: *The TVQVC architecture. The encoder of TVQVC is modified from Conformer, and the decoder of TVQVC is modified from FastSpeech decoder.*

codes. And because of using global codebook, the embeddings after vector quantization can be seen without speaker information. The baseline model can be divided into three modules. The encoder takes the fbanks as inputs, which are downsampled by a factor of 2 and processed by several convolution blocks. The bottleneck layer projects contiguous embeddings into discretized codes. After time jitter on quantized codes, the decoder uses an autoregressive decoder with teacher forcing to reconstruct the origin waveform.

2.2. TVQVC

The proposed method is illustrated in Figure 2, which can be decomposed into three parts: encoder, bottleneck layer and decoder. Each part will be introduced in the following subsections. The whole structure of proposed model is an autoencoder based on reconstructed training, which doesn't require parallel training data.

2.2.1. Conformer Encoder

Conformer[27] is a convolution-augmented transformer for speech recognition, which is composed of four modules stacked together, a feed-forward module, a self-attention module, a convolution module and a second feed-forward module. The convolution module consistes of a pointwise convolution and a depthwise convolution. For input x_i of conformer block i, the output y_i of this block is:

$$\begin{aligned} \tilde{x_i} &= x_i + \frac{1}{2}\mathrm{FFN}(x_i) \\ x_i^{'} &= \tilde{x}^i + \mathrm{MHSA}(\tilde{x}_t^i) \\ x_i^{''} &= x_i^{'} + \mathrm{Conv}(x_i^{'}) \\ y_i &= \mathrm{Layernorm}(x_i^{''} + \frac{1}{2}\mathrm{FFN}(\mathrm{x}_\mathrm{i}^{''})) \end{aligned} \tag{1}$$

where MHSA refers to Multi-Head Self-Attention module, Conv refers to the convolution module and FFN refers to the Feed forward module.

2.2.2. Vector-quantized module

The vector-quantized layer consists of a trainable codebook $\{e_1, e_2, ..., e_K\}$ with K distinct codes. While in the forward process, a sequence of contiguous feature vectors $y := \langle y_1, y_2, ..., y_T \rangle$ are discretized by mapping each z_i to it's nearest neighbour of $L2\text{-}norm$ in the codebook. We can find $k := \arg\min_n \|y_i - e_n\|$. But the operator of $argmin$ is not differentiable. Origin vector-quantized layer loss should be:

$$L_{vq} = \|\mathrm{sg}\,[z_e(y)] - e\|_2^2 + \|z_e(y) - \mathrm{sg}\,[e]\,\|_2^2 \tag{2}$$

Where sg stands for the stop-gradient operator, y represents input sequences of fbank and z_e stands for encoder. But here we use $exponential\ moving\ averages$ for replacing $\|z_e(y) - \mathrm{sg}\,[e]\,\|_2^2$ to update the dictionary items[28]. Finally, a $commitment\ cost$ is added to the vector-quantized loss to encourage each z_i to commit to the selected code.

2.2.3. Jitter module

We expect the encoder to learn high dimensional representations which correspond to phonetic content. In other words, one embedding of vector-quantized layer can represent a special phonetic content. We use temporal jitter regularization to prevent the latent vector co-adaptation and reduce the model sensitivity near the unit boundary that during training each latent vector can replace either one or both it's neighbors[25]. We could apply the jitter by:

$$j_t \sim Categorical(p, 1 - 2 * p, p) \quad j_t \in \{-1, 0, 1\} \tag{3}$$

$$\hat{c}_t = \begin{cases} c_{t-1}, & \text{if } j_t = -1 \text{ and } t > 1 \\ c_{t+1}, & \text{if } j_t = 1 \text{ and } t < T \\ c_t, & \text{else} \end{cases} \quad (4)$$

where p is the jitter probability and $p \in [0, 0.5]$, c_t is the output of vector quantization layer at time t and $\hat{c}_t$ is the new assigned codebook index after jitter.

2.2.4. Fundamental frequency module

Due to vector-quantized layer, we found that there are some problems of the fundamental frequency in converted utterances. So we followed the method in [15]. In addition to speaker embedding s, we conditioned the decoder of TVQVC with a per-frame feature p_n, which directly computed from the source speaker's F0. Firstly, we extracted the log-F0 of source speaker's voice samples using $pyin$ and compute the log-F0's mean μ and variance σ^2 of per speaker. Then the input speech's log-F0 p_{src} was normalized by $p_{norm} = (p_{src} - u)/\sigma/4$. Then we limited p_{norm} within 0 and 1 and quantized the range 0-1 into 257 dim one-hot embedding. And we also added another bin to represent unvoiced or voiced flag.

2.2.5. Implementation details

We adopted the conformer module from the transformer based hybird CTC/attention ASR model in espnet implementation[2] as our method's encoder. But different from the origin implementation, we modified the subsampling module from $1/4$ to $1/2$, the kernel size of convolution module is 31, the attention dim of encoder is 320 and the channel of input fbank is 80. One-hot vector is used for speaker identity representation and goes through a speaker embedding layer to get a 80-dimensional speaker embedding. The implementation of CTC loss was modified from espnet and we used mono-phone-level CTC loss instead of character-level CTC loss. For vector-quantized layer, the baseline and our proposed method both have 512 320-dimensional codes in codebook. The jitter probability is set to 0.2 for both baseline and this method. The normalized quantized fundamental frequency and voiced flag are combined together to get 258-dimensional embedding. Then they are concatenated together across the time before fed into decoder. Training loss L_{train} is composed of vq loss L_{vq}, CTC loss L_{ctc} and reconstruct loss L_{recon}. In the trainging phase, the reconstruct loss can be written as Eq. 5.

$$L_{recon} = E[\|X - \hat{X}\|_2^2] \quad (5)$$

Where X represents the input mel-spectrum and $\hat{X}$ represents the reconstructed mel-spectrum. Training loss L_{train} can be written below:

$$L_{train} = \lambda_1 L_{recon} + \lambda_2 L_{vq} + \lambda_3 L_{ctc} \quad (6)$$

where we set $\lambda_1 = 1.0$, $\lambda_2 = 5.0$ and $\lambda_3 = 0.05$.

3. Experimental Setup

3.1. Datasets

Our experiments were conducted on the open source mandarin datasets, including 85h mandarin multi-speaker dataset AISHELL3[29], 12h mandarin female dataset databaker[3], 5h male and 5h female M2VoC dev set[4]. Among them, 5h male and 5h female M2VoC dev set is parallel data. We converted the audio from 48000Hz into 16000Hz and removed the speakers of less than 400 utterances. We randomly picked 200 utterances of baker, M2VoC male and female for evaluation respectively.

3.2. Training details

We trained the proposed model by using Adam optimizer with learning rate of 0.0004, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and batchsize of 30. And all models were trained up to 200k iterations. The mel-spectrum and F0 shared the same parameters, whose win length is 640, hop length is 160 and FFT length is 1024. Different from baseline VQVAE which directly generates waveform. We used HiFiGAN[5] to convert the mel-spectrum to waveform[30], which was trained on databaker, AISHELL3 and M2VoC dev set. We also added a new baseline model PPGVC, whose ASR module was trained on our own training data of telephone channel about 10k hours, the acoustic model of converting PPGs to mel-spectrum is modified from Tacotron2[31].

4. Experiments

4.1. Subjective evaluations

In our experiments, Subjective evaluation in terms of naturalness and speaker similarity of converted speech were conducted. There were at least twenty listeners involved, they were asked to give a 5-scale opinion score(5: excellent, 4:good, 3: fair, 2: poor, 1: bad) on both speaker similarity and naturalness for each converted utterances, which were selected randomly from the evaluation set.

Table 1: *Table of Naturalness and Speaker similarity*

Method	Naturalness	Speaker similarity
Baseline	2.97 ± 0.08	3.56 ± 0.12
PPGVC	3.56 ± 0.10	3.33 ± 0.07
Proposed	$\mathbf{3.78 \pm 0.09}$	$\mathbf{3.72 \pm 0.07}$

From Table 1, we can find that our proposed model performs better than VQVAE and PPGVC both on naturalness and speaker similarity.

4.2. Objective evaluations

Mel Cepstral Distortion (MCD) was used for the objective evaluation of voice conversion. While a converted utterance doesn't align with corresponding parallel utterance entirely, so we applied dynamic time warping to align the parallel and converted utterances.

$$\text{MCD[dB]} = (10/ln10)\sqrt{2\sum_{i=1}^{K}(MCC_i^t - MCC_i^c)^2} \quad (7)$$

where MCC represents mel-cepstral coefficient, K is the dimension of the MCCs, and MCC_d^c and MCC_d^t represents the d-th dimensional coefficient of the converted MCCs and the target MCCs respectively. We use pysptk tool to extract the MCCs.

[2]https://github.com/espnet/espnet

[3]https://www.data-baker.com/#/data/index/source

[4]http://challenge.ai.iqiyi.com/M2VoC

[5]https://github.com/jik876/hifi-gan

To evaluate the F0-RMSE, we also used dynamic time warping to align pair of f0 sequences. The computation of F0-RMSE is as:

$$\text{F0-RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}((F0_i^t) - (F0_i^c))^2} \quad (8)$$

Where N is the number of frames. F0_i^c and F0_i^t are F0 value at the i-th frame of the converted speech and target speech respectively.

A speaker verification system[6] was also adopted for objective evaluation of the converted speech. The SV system took a converted utterance as input and generated a fix-length embedding. We calculated the cosine distance between the embeddings of converted utterances and real utterances to evaluate the similarity objectively.

Table 2: *Table of MCD , F0-RMSE and D-vector distance. Lower is better.*

Method	F0-RMSE (Hz)	MCD (dB)	D-vector distance
VQVAE	20.99	2.69	0.1094
PPGVC	**17.87**	2.30	0.1009
Proposed	18.77	**2.26**	**0.0912**

The objective evaluation results of conversion between male and female of M2VoC were shown in Table 2 and we can see that the proposed method achieves best performance on MCD and D-vector distance. And PPGVC is slightly better than the proposed method on F0-RMSE.

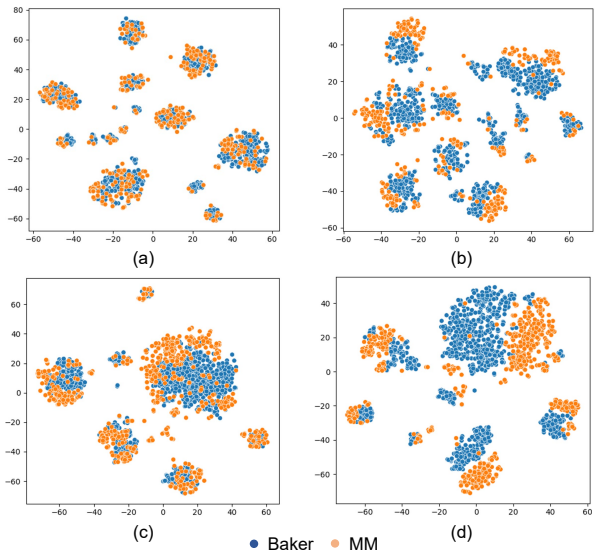

Figure 3: *Compared with and without vector-quantized layer. Blue and orange points represent the embeddings of baker and M2VoC dev male respectively.*

[6]https://github.com/resemble-ai/Resemblyzer

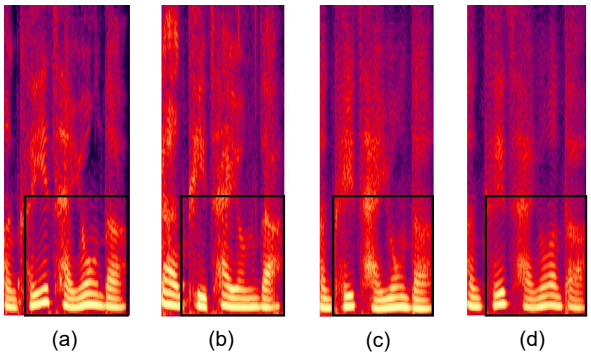

Figure 4: *The difference of generated frequency spectrum. (a) represents real spectrum, (b) represents the VQVAE model, (c) represents the proposed model without quantization f0 and (d) represents the proposed model.*

4.3. Ablation experiment

To prove that with vq layer is helpful for removing residual speaker information. We compared the proposed model with vq layer with without vq layer. Firstly, we trained them up to about 200k steps. Then we took CTC force-align on the target labels and hidden embeddings of encoder to get the specific labels of hidden embeddings. We picked label $a4$ and $ei3$ for testing and verifying our hypothesis. As shown in Fig.3, orange and blue points represent baker and M2VoC male respectively. (a) and (c) are extracted by proposed model, (b) and (d) are extracted by proposed model without vq layer. (a) and (b) have same phone label, the same as (c) and (d). We pick the same 2000 points of two methods equally and use tsne to reduce 320-dim embeddings to 2-dim. Obviously, we can observe that (b) and (d) have more speaker differentiation while in (a) and (c) embeddings fuse together. So the vq layer can force encoder to remove more speaker information.

To validate the efficiency of quantizated fundamental frequency, we compared the detail of conversion utterances and real utterances. In the Figure 4, the shape of hump resonance is the most different, which causes it sounded strangely. And the detail of frequency spectrum without guided normalized f0 is not as rich as the proposed model.

We also found that if there is no CTC loss, the network can't converge well and $L1\text{-}norm$ loss is about always maintain at about 1.0 while proposed at about 0.2. The generated audio can be found in our demo page[7].

5. Conclusions

In this paper, we present a novel structure of combining transformer and VQVAE with CTC loss and reconstruction loss to achieve high-quality voice conversion. The subjective and objective results show that the converted speech's naturalness and speaker similarity of proposed model is better than VQVAE and PPGVC. And ablation experiments also show the efficiency of combining CTC loss with vector quantization and quantization normalized fundamental frequency, which can improve the speaker similarity and naturalness respectively.

[7]https://miracyan.github.io/TVQVC/

6. References

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

[2] T. Toda, A. W. Black, and K. Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 8, pp. 2222–2235, 2007.

[3] D. Erro, A. Moreno, and A. Bonafonte, "Voice conversion based on weighted frequency warping," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 5, pp. 922–931, 2009.

[4] T. Toda, H. Saruwatari, and K. Shikano, "Voice conversion algorithm based on gaussian mixture model with dynamic frequency warping of straight spectrum," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)*, vol. 2. IEEE, 2001, pp. 841–844.

[5] L. Sun, S. Kang, K. Li, and H. Meng, "Voice conversion using deep bidirectional long short-term memory based recurrent neural networks," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 4869–4873.

[6] L. Sun, K. Li, H. Wang, S. Kang, and H. Meng, "Phonetic posteriorgrams for many-to-one voice conversion without parallel data training," in *2016 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2016, pp. 1–6.

[7] S. Liu, J. Zhong, L. Sun, X. Wu, X. Liu, and H. Meng, "Voice conversion across arbitrary speakers based on a single target-speaker utterance." in *Interspeech*, 2018, pp. 496–500.

[8] J. Serrà, S. Pascual, and C. Segura, "Blow: a single-scale hyperconditioned flow for non-parallel raw-audio voice conversion," *arXiv preprint arXiv:1906.00794*, 2019.

[9] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *arXiv preprint arXiv:1406.2661*, 2014.

[10] T. Kaneko and H. Kameoka, "Cyclegan-vc: Non-parallel voice conversion using cycle-consistent adversarial networks," in *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, 2018, pp. 2100–2104.

[11] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "Cyclegan-vc2: Improved cyclegan-based non-parallel voice conversion," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6820–6824.

[12] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "Stargan-vc: Non-parallel many-to-many voice conversion using star generative adversarial networks," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 266–273.

[13] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "Stargan-vc2: Rethinking conditional methods for stargan-based voice conversion," *arXiv preprint arXiv:1907.12279*, 2019.

[14] K. Qian, Y. Zhang, S. Chang, X. Yang, and M. Hasegawa-Johnson, "Autovc: Zero-shot voice style transfer with only autoencoder loss," in *International Conference on Machine Learning*. PMLR, 2019, pp. 5210–5219.

[15] K. Qian, Z. Jin, M. Hasegawa-Johnson, and G. J. Mysore, "F0-consistent many-to-many non-parallel voice conversion via conditional autoencoder," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6284–6288.

[16] D.-Y. Wu and H.-y. Lee, "One-shot voice conversion by vector quantization," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7734–7738.

[17] 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.

[18] C.-C. Hsu, H.-T. Hwang, Y.-C. Wu, Y. Tsao, and H.-M. Wang, "Voice conversion from non-parallel corpora using variational auto-encoder," in *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*. IEEE, 2016, pp. 1–6.

[19] Y. Zhou, X. Tian, H. Xu, R. K. Das, and H. Li, "Cross-lingual voice conversion with bilingual phonetic posteriorgram and average modeling," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6790–6794.

[20] S. Liu, Y. Cao, S. Kang, N. Hu, X. Liu, D. Su, D. Yu, and H. Meng, "Transferring source style in non-parallel voice conversion," *arXiv preprint arXiv:2005.09178*, 2020.

[21] J.-X. Zhang, Z.-H. Ling, and L.-R. Dai, "Recognition-synthesis based non-parallel voice conversion with adversarial learning," *arXiv preprint arXiv:2008.02371*, 2020.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017.

[23] S. Liu, Y. Cao, D. Wang, X. Wu, X. Liu, and H. Meng, "Any-to-many voice conversion with location-relative sequence-to-sequence modeling," *arXiv preprint arXiv:2009.02725*, 2020.

[24] B. van Niekerk, L. Nortje, and H. Kamper, "Vector-quantized neural networks for acoustic unit discovery in the zerospeech 2020 challenge," *arXiv preprint arXiv:2005.09409*, 2020.

[25] J. Chorowski, R. J. Weiss, S. Bengio, and A. 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.

[26] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[27] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.

[28] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, "Neural discrete representation learning," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 6309–6318.

[29] Y. Shi, H. Bu, X. Xu, S. Zhang, and M. Li, "Aishell-3: A multi-speaker mandarin tts corpus and the baselines," *arXiv preprint arXiv:2010.11567*, 2020.

[30] 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, 2020.

[31] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.