FlowCPCVC: A Contrastive Predictive Coding Supervised Flow Framework for Any-to-Any Voice Conversion

Jiahong Huang, Wen Xu, Yule Li, Junshi Liu, Dongpeng Ma,Wei Xiang

Bigo Technology PTE. LTD, Singapore

{huangjiahong.dracu,xuwen,liyule,liujunshi,madongpeng,xiangwei1}@bigo.sg

Abstract

Recently, the research of any-to-any voice conversion(VC) has been developed rapidly. However, they often suffer from unsatisfactory quality and require two stages for training, in which a spectrum generation process is indispensable. In this paper, we propose the FlowCPCVC system, which results in higher speech naturalness and timbre similarity. FlowCPCVC is the first one-stage training system for any-to-any task in our knowledge by taking advantage of VAE and contrastive learning. We employ a speaker encoder to extract timbre information, and a contrastive predictive coding(CPC) based content extractor to guide the flow module to discard the timbre and keeping the linguistic information. Our method directly incorporates the vocoder into the training, thus avoiding the loss of spectral information as in two-stage training. With a fancy method in training any-to-any task, we can also get robust results when using it in any-to-many conversion. Experiments show that FlowCPCVC achieves obvious improvement when compared to VQMIVC which is current state-of-the-art any-to-any voice conversion system. Our demo is available online [1].

Index Terms: flow, voice conversion, contrastive predictive coding, vector quantization, generative adversarial network

1. Introduction

Voice conversion is a technique for converting one speaker's voice identity into another one while preserving the linguistic content. The any-to-any(one-shot) task for VC is to convert any voice to any target speaker even unseen during training. The converted voice should preserve target speaker identity [1], source audio prosody [2] and accent [3]. The research of voice conversion is becoming more and more popular owing to its high potential for various applications, such as speaking aids [4, 5] and style[6, 7] and pronunciation [8] conversion.

There are many works trying to decouple the linguistic content from audio with unsupervised learning, such as Auto-encoder-based approaches [9, 10, 11, 12] or GAN-based methods [13, 14]. However, they don't have an explicit linguistic monitor module to guide the training process to disentangle linguistic and timbre information, which leads to degradation of VC performance. TTS-based approaches, such as [15, 16, 17], require explicit text labels or phone posteriorgram(PPG) features to monitor the model to extract linguistic content information. However, text labels are not often available at hand and content information will be lost if the pre-trained ASR model is not robust. In order to solve the problems mentioned above, some approaches FragmentVC [18], VQMIVC [19] and [20] adopt an unsupervised linguistic content learning module to extract content information. But their training and inference rely entirely on the output of unsupervised module, this will mix with information other than the content information. Some any-to-any methods like [19, 21] extract timbre information by using internal learning modules, but it will get poor generalization performance when training with small data and insufficient similarity to target timbre that was unseen from training. Another point is that the previous approaches require vocoders trained or fine-tuned with first-stage model output, which causes training and deployment inefficiency. What's more, hidden representations will be lost when using predefined intermediate features, such as mel-spectrogram.

In order to avoid content information mixing with other information, we restrict the output of CPC module to follow a Gaussian distribution and adopt Flow network to fit it, rather than rely entirely on the output of CPC. To improve timbre similarity and generalization, we use an external pre-trained speaker-independent encoder [22] for extracting timbre information. Additionally, we directly incorporate the vocoder into the voice conversion network to form a unified network, so that the entire training requires only one stage. Finally, we put forward a new framework, FlowCPCVC, which is a variational autoencoder(VAE) with generative adversarial learning for training. It includes five modules: timbre encoder, posterior encoder, Flow, CPC-Net and decoder. Our paper has three main contributions: 1) We proposed a novel framework for high-quality any-to-any voice conversion. 2) The first single stage training system, as far as we know, that combines vocoder into conversion network. 3) A fancy way to implement any-to-many task while training the any-to-any task.

The rest of this paper is organized as follows: Section 2 presents the proposed FlowCPCVC system. The details of training for any-to-many task will be shown in Section 3. Finally, the experiments are described in Section 4.

2. Proposed approach

This section first describes the system architecture of the FlowCPCVC, then elaborates on the knowledge to learn the linguistic content with CPC-Net, and finally shows on the detail for training.

2.1. Architecture of the FlowCPCVC system

Our framework of FlowCPCVC was inspired by a text-to-speech framework [23]. As shown in Figure 1, FlowCPCVC mainly includes five modules: 1) Timbre encoder, the main function of this module is extracting timbre information for training. 2) Posterior encoder module, force to find the implicit representation of linear spectrogram which is sent to decoder to reconstruct the waveform. 3) Flow module, aim to remove the timbre information and keep the linguistic information. 4) CPC-Net module, which is a framework for extracting the linguistic information through unsupervised comparative learning

[1] https://aijianiula0601.github.io/FlowCPCVC

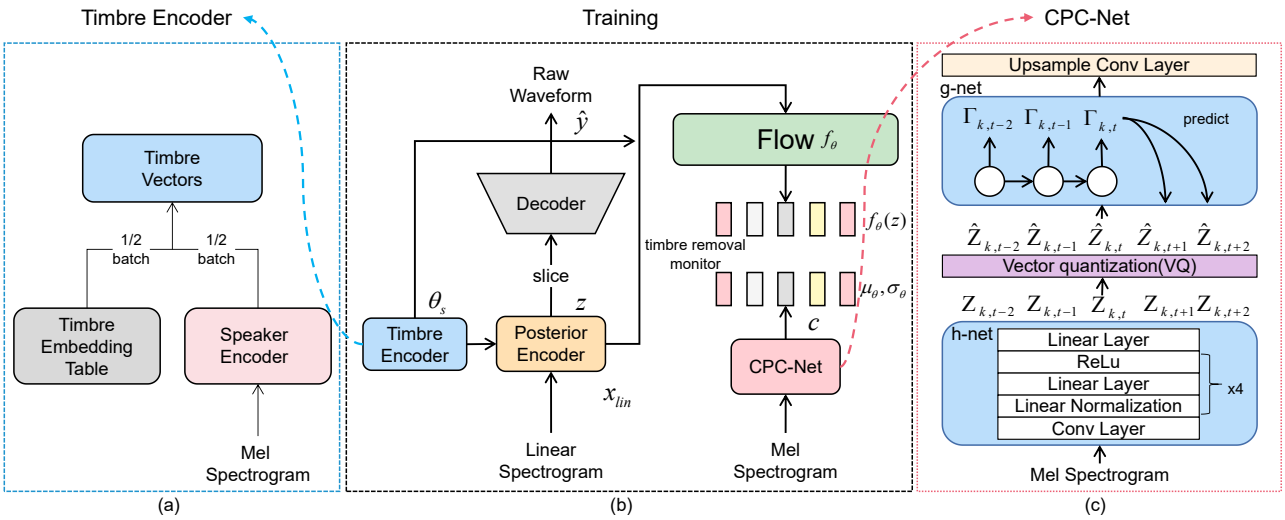

Figure 1: *The architecture of FlowCPCVC. (a) Timbre vectors from speaker encoder or timbre emebedding table. (b) Overall procedures for training. (c) Architectural components of CPC-Net.*

[20, 24], aim to extract linguistic information from audio and monitor the output of flow module to represent linguistic information. 5) Decoder module, which inherit from hifigan [25].

FlowCPCVC can be expressed as conditional VAE, which is a generative model in the form of $p_\theta(x|c) = \int p(z)p_\theta(x|z)dz$, where c is for linguistic information from CPC-Net, $p(z)$ is a prior distribution over latent variables z and $p_\theta(x|z)$ is the likelihood function of a data point x given latent variables z which can be considered as a decoder. It is parameterized by a neural network θ. Since the true posterior $p_\theta(z|x)$ over the latent variables of a VAE is usually analytically intractable, we approximate it with a variational distribution $q_\phi(z|x)$, which can be viewed as an encoder. Parameters θ and ϕ can be optimized by maximizing the variational lower bound, also called the evidence lower bound (ELBO), of the intractable marginal log-likelihood of data log $p_\theta(x|c)$:

$$\begin{aligned} logp_\theta(x|c) &\geq \mathbf{E}_{q_\theta(z|x)}[log\frac{p_\theta(x,z)}{q_\phi(z|x)}] \\ &= \mathbf{E}_{z\sim q_\theta(z|x)}[logp_\theta(x|z) - log\frac{q_\phi(z|x)}{p_\theta(z)}] \\ &= \mathbf{E}_{z\sim q_\theta(z|x)}[logp_{\theta(x|z)}] - \mathbf{KL}(q_\theta(z|x)||p_\theta(z)) \end{aligned} \tag{1}$$

The training loss is then the negative ELBO, which can be viewed as the sum of reconstruction loss $L_{recon} = -logp_\theta(x|z)$ and KL divergence loss $L_{kl} = logq_\phi(z|x) - logp_\theta(z)$, where $z \sim q_\phi(z|x) = N(z; \mu_\phi(x, \sigma(x)))$, $x = x_{lin}$.

Timbre Encoder θ_s: 256-dim timbre vectors come from Speaker Encoder or Timbre Embedding Table. We extract the timbre vector from Speaker Encoder for any-to-any task. The architecture of Speaker Encoder comes from voxceleb-trainer [22], which is used for the task of independent speaker verification. We choose the ResNetSE34V2 net for our task and trained the model with 30 thousand speakers which include Chinese, English and Bengal. The EER of pretrained model in test dataset is 0.2%.

Posterior Encoder $\theta(x)$: The posterior Encoder compresses the linear spectrogram to implicit representation z, which approximately equivalent to posterior distribution $p_\theta(z|x)$. The output of posterior encoder is send to decoder and flow.

Flow $\theta(z)$: To increase the expressiveness of the prior distribution for generating the realistic samples, which has been proved in [23], and better to fit the implicit phonetic information of CPC output, we apply the normalizing flow f_θ [26], which allows an invertible transformation of a simple distribution into a more complex distribution following the rule of change-of-variables, on top of the factorized normal prior distribution:

$$p_\theta(z|c) = N(f_\theta(z); \mu_\theta(c); \sigma_\theta(c))|det\frac{\partial f_\theta(z)}{\partial_z}| \tag{2}$$

where c is content repressetation extract from cpc-Net.

CPC-Net θ_c: The task of contrastive predictive coding(CPC) network, shortened to CPC-Net, is to extract the linguistic content from speech. As shown in figure 1 part c. The CPC-net includes h-net,VQ [20] and g-net modules. We use the mel-spectrogram as acoustic feature and randomly select T frames from each utterance for training. The k_{th} frame is denoted as $X_k = X_{k,1}, X_{k,2}...X_{k,T}$. The h-net takes in X_k to get Z_k, the VQ quantifies the Z_k to $\hat{Z}_k$ and the g-net transforms $\hat{Z}_k$ to $\hat{R}_k$. Finally, we use an upsample Conv layer to turn $\hat{R}_k$ to Gaussian distribution, which is expressive to phonetic representation. This is inspired by the use of GMM to model phonemic information in speech recognition.

h-net: The h-net contains 2 layers and one block which repeated 4 times. It first uses one Conv1d Layer with a stride of 2 to reduce the T frames to T/2 frames, then sends them to four repeated blocks, which contains layer normalization, 512-dim linear layer, and the Relu function as activation. The outputs Z_k of blocks are sent to a linear layer.

VQ-operation: VQ [20] operation discretizes the Z_k with a trainable codebook B, which has 512 64-dim learnable vectors, into $\hat{Z}_k = \{\hat{Z}_{k,1}, \hat{Z}_{k,2}, ..., \hat{Z}_{k,T/2}\}$, where $\hat{z}_{k,t} \in B$ is vector closest to $z_{k,t}$. It learns representations that remove non-essential details in Z_k, making $\hat{Z}_k$ to be related with underlying linguistic information. The loss of VQ [20]:

$$L_{VQ} = \frac{2}{KT}\sum_{k=1}^{K}\sum_{t=1}^{T/2}\|z_{k,t} - sg(\hat{z}_{k,t})\|_2^2 \tag{3}$$

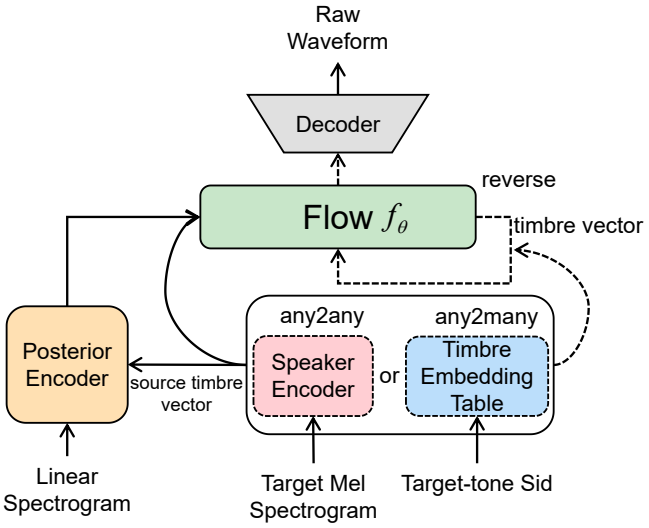

Figure 2: *Inference of FlowCPCVC.*

where sg(.) denotes the stop-gradient operator.

g-net: The g-net is a 256-dim unidirectional RNN layer, which encourages the $\hat{Z}_k$ to capture local structures $R_k = \{r_{k,1}, r_{k,2}, ..., r_{k,T/2}\}$. Given the $r_{k,t}$, the model is trained to distinguish a positive sample $\hat{z}_{k,t+m}$ that $m = 6$ from negative samples drawn from the set $\Omega_{k,t,m} = 10$ in the future by minimizing the InfoNCE loss [27]:

$$L_{CPC} = -\frac{1}{KT'M}\sum_{k=1}^{K}\sum_{t=1}^{T'}\sum_{m=1}^{M} log[\frac{exp(\hat{z}_{k,t+m}^{T} W_m r_{k,t})}{\sum_{\tilde{z}\in\Omega_{k,t,m}} exp(\tilde{z}^T W_m r_{k,t})}] \tag{4}$$

where $T' = T/2 - M, W_m (m = 1, 2, ...M)$ is trainable projection matrix. The negative samples are selected from the same utterance as positive samples. Finally, a transposed Conv1d layer upsamples the outputs of g-net to T frames and calculates the mean $\mu_\theta(c)$ and variance $\sigma_\theta(c)$ of each frame.

Decoder: The decoder comes from HiFi-GAN[25]. We use mel-spectrogram instead of a raw waveform, denoted by x_{mel}, to caculate the reconstruction loss for target data. We upsample the latent variables z to the waveform domain $\hat{y}$ through a decoder and transform $\hat{y}$ to the mel-spectrogram domain $\hat{x}_{mel}$. Then the L1-norm between the predicted and target is used as the reconstruction loss:

$$L_{recon} = \|x_{mel} - \hat{x}_{mel}\|_1 \tag{5}$$

This can be viewed as maximum likelihood estimation assuming a Laplace distribution for data distribution and ignoring constant terms. We define the reconstruction loss in the mel-spectrogram domain to improve the perceptual quality by using a mel-scale that approximates the response of the human auditory system. Note that the mel-spectrogram estimation from a raw waveform does not require trainable parameters as it only uses STFT and linear projection onto the mel-scale. Furthermore, the estimation is only employed during training, not inference. In practice, we do not upsample the whole latent variables z but use partial sequences as an input for the decoder, which is the windowed generator training used for efficient end-to-end training [23].

2.2. Training objectives

We trained the model as shown in figure 1. Similar to the learning TTS system [23], we adopted adversarial training in our learning system. The HiFi-GAN [25] generator was used as a decoder, which takes the outputs of Posterior Encoder as input. A discriminator D is added to distinguish between the output generated by the decoder G and the ground truth waveform y, while the generator tries to fool the discriminator by generating the predicted speech $\hat{y}$ that similar to the real speech y. Two types of losses are successfully applied to the decoder; the least squares loss function [28] for adversarial training, and the additional feature matching loss [29, 30] for training the generator:

$$\begin{aligned} L_{adv}(D) &= E_(y,z) = [(D(y)-1)^2 + (D(G(z))^2)] \\ L_{adv}(G) &= E_z[(D(G(z))-1)^2] \\ L_{fm}(G) &= E_{(y,z)}[\sum_{l=1}^{T}\frac{1}{N_l} \parallel D^l(y) - D^l(G(z)) \parallel_1] \end{aligned} \tag{6}$$

where T denotes the total number of layers in the discriminator and D^l outputs the feature map of the l-th layer of the discriminator with N_l number of features. It is important to note that feature matching loss can be regarded as reconstruction loss measured in the hidden layer of the discriminator and seen as an alternative to the element reconstruction loss for VAE [29].
The total loss for training our conditional VAE can be expressed as follows:

$$L_{vae} = L_{recon} + L_{kl} + L_{adv}(G) + L_{fm}(G) + L_{VQ} + L_{CPC} \tag{7}$$

3. Methods for any-to-many task

Although any-to-any task has covered the efficacy of any-to-many, we still want to implement the any-to-many task, that the information of timbre learned from various audios for one speaker is more stable and robust. As shown in figure 1, in order to combine any-to-any and any-to-many tasks for simultaneous training, we created a fixed-size embedding table in the model to learn a stable timbre information for target speakers. During the training, a batch of timbre vectors is divided into two parts, of which one half is obtained from the Timbre Embedding Table, while another half is extracted from the Speaker Encoder. Merged into a batch and sent to Posterior Encoder and Flow modules.

In the phase of inference, as shown in Figure 2, if any-to-any task is required, the timbre vectors for source and target are obtained from the Speaker Encoder. If we need to do any-to-many tasks, the timbre vector for the source audio is retrieved from the Speaker Encoder, while the target timbre that we want to achieve is retrieved from the Timbre Embedding Table. The whole process of inference is as follows: Both the linear spectrum and the source timbre vector are fed to the posterior encoder, and the output of posterior encoder is sent to the flow module. The output of flow is sent to the reversed flow module together with the target timbre vector, and its output is sent to the decoder. Finally, we will get the audio of the target timbre.

4. Experiments

4.1. Datasets

We conducted experiments on VCTK dataset [31] with 110 English speakers. We selected 90 speakers for training and 20 for

testing following the setting of VQMIVC [19]. To verify the effect of one-shot VC, we treated testing speakers as unseen speakers. The audio's format of VCTK is 16-bit PCM with sample rate of 44 kHz. We downsampled the audio to 22 kHz.

4.2. Preprocessing

The linear spectrograms can be obtained from raw waveforms through the Short-time Fourier transform (STFT), used as input of the posterior encoder for acoustic features extraction. We used 80- dimensional mel-scale spectrograms for reconstruction loss and as input of CPC-Net. We set FFT size, window size and hop size of the transform to 1024, 1024 and 256.

4.3. Training

For the FlowCPCVC network training, we use Adam optimizer[10] with $\beta_1 = 0.8$, $\beta_2 = 0.99$ and weight decay $\lambda = 0.01$. The learning rate decay is scheduled by a $0.999^{1/8}$ factor in every epoch with an initial learning rate of 2×10^{-4}. Following previous work [32], we adopt the windowed generator training, a method of generating only a part of raw waveforms to reduce the time and memory usage during training. We randomly extract segments of latent representations with a window size of 32 to feed to the decoder instead of feeding entire latent representations and also extract the corresponding audio segments from the ground truth raw waveforms as training targets. We use mixed precision training on 4 NVIDIA V100 GPUs. The batch size is set to 40 per GPU and the model is trained up to 82k steps. We compare our method with FragmentVC [21], MediumVC [33] and VQMIVC [19] where all models were trained and tested under the same data partitioning rules.

4.4. Experimental results and analysis

4.4.1. Content preservation and F0 variation consistency

To evaluate the retention degree of linguistic information and intonation information from original audio, we tested the CER and WER of converted audio with the publicly released Wenet ASR system [34]. To evaluate the similarity of intonation, we extract the f0 of audio and calculate the Pearson correlation coefficient(PCC) between the source and converted audio. The higher F0-PCC denotes that the converted voice has better similarity of intonation to source voice. For each speaker in the test dataset, we randomly select one audio, half of the twenty audios are used as the source, and half are used as the target, then we get 100 converted pairs after pairwise combination. The results for different methods are shown in Table 1 that our system is better in retention of linguistic content and more similar to source voice in intonation with higher PCC value.

Table 1: *ASR and F0-PCC results for one-shot VC*

Methods	CER	WER	F0-PCC
origin	3.5%	9.0%	1.0
FragmentVC	86.6%	91.5%	0.231
MediumVC	15.2%	30.3%	0.695
VQMIVC	14.2%	29.1%	0.792
our	**13.8%**	**28.4%**	**0.870**

4.4.2. Speech naturalness and Timbre similarity

In order to measure the naturalness and timbre similarity, a comparative test was designed, and the MOS score of the converted audio was performed manually. We asked 20 native English speakers to rate the audios. The MOS has 5 levels: 1-bad, 2-poor, 3-fair, 4-Good, 5-excellent. We separated the data into different genders in the test dataset, randomly selected 5 clips for each speaker, and tested in female to male(F2M), male to female(M2F), female to female(F2F) and Male to male (M2M). The results are shown in Table 2, which show that our method has advantages in both naturalness and timbre similarity.

Table 2: *The MOS of Speech naturalness and speaker similarity*

Method	MOS				SIM
	F2M	M2F	F2F	M2M	
Ground truth		4.66			-
FragmentVC	1.41	1.62	1.81	1.76	2.03
MediumVC	3.25	3.11	3.41	3.51	3.05
VQMIVC	3.42	3.23	3.74	3.85	3.81
our	**4.08**	**4.11**	**4.28**	**4.23**	**4.24**

4.4.3. Contrast of emotional conversion effect

A good voice conversion system can not only change the timbre of common pronunciation, but also the emotional voice, such as ah, um, loud cry and so on. In order to verify the conversion effect of our voice conversion system for such unusual pronunciation, we specially selected 50 audio tones containing mood sounds for conversion test. Compared with other systems, we use the same grades for MOS as in Experiment 2. The comparison result is shown in Table 3. We can see that our method has a better conversion effect for such voice.

Table 3: *The MOS of Speech naturalness and speaker similarity for emotional audios*

Method	MOS	SIM
Ground truth	4.64 ± 0.8	-
FragmentVC	1.81 ± 0.10	1.54 ± 0.11
MediumVC	2.20 ± 0.12	2.04 ± 0.12
VQMIVC	2.94 ± 0.15	3.34 ± 0.14
our	$\mathbf{3.57} \pm 0.11$	$\mathbf{3.63} \pm 0.09$

5. Conclusion

In this paper, we proposed a new framework for voice conversion system, which we call FlowCPCVC. In order to preserve the source linguistic information and improve naturalness, we adopted Gaussian distribution of CPC-Net outputs to monitor the flow module to remove timbre and preserve the linguistic information to generate more natural audio. To increase timbre similarity, we introduced a robust speaker-independent encoder to assist training. Additionally, our model is the first single stage training voice conversion system for one-shot task, unlike other models that require separated training of acoustic models and vocoder. In the future, we will try to design a timbre extraction model due to that ResNetSE34V2 is designed for speaker recognition rather than timbre extraction, or remove the Speaker Encoder for any-to-many task to create a more lightweight VC network.

6. References

[1] S. H. Mohammadi and A. Kain, "An overview of voice conversion systems," *Speech Communication*, vol. 88, pp. 65–82, 2017.

[2] D. Rentzos, S. Vaseghi, E. Turajlic, Q. Yan, and C.-H. Ho, "Transformation of speaker characteristics for voice conversion," in *2003 IEEE Workshop on Automatic Speech Recognition and Understanding (IEEE Cat. No. 03EX721)*. IEEE, 2003, pp. 706–711.

[3] K. Oyamada, H. Kameoka, T. Kaneko, H. Ando, K. Hiramatsu, and K. Kashino, "Non-native speech conversion with consistency-aware recursive network and generative adversarial network," in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2017, pp. 182–188.

[4] A. B. Kain, J.-P. Hosom, X. Niu, J. P. Van Santen, M. Fried-Oken, and J. Staehely, "Improving the intelligibility of dysarthric speech," *Speech communication*, vol. 49, no. 9, pp. 743–759, 2007.

[5] K. Nakamura, T. Toda, H. Saruwatari, and K. Shikano, "Speaking-aid systems using gmm-based voice conversion for electrolaryngeal speech," *Speech communication*, vol. 54, no. 1, pp. 134–146, 2012.

[6] Z. Inanoglu and S. Young, "Data-driven emotion conversion in spoken english," *Speech Communication*, vol. 51, no. 3, pp. 268–283, 2009.

[7] T. Toda, M. Nakagiri, and K. Shikano, "Statistical voice conversion techniques for body-conducted unvoiced speech enhancement," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 9, pp. 2505–2517, 2012.

[8] T. Kaneko, H. Kameoka, K. Hiramatsu, and K. Kashino, "Sequence-to-sequence voice conversion with similarity metric learned using generative adversarial networks." in *INTERSPEECH*, vol. 2017, 2017, pp. 1283–1287.

[9] 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.

[10] 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.

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

[12] W.-C. Huang, H. Luo, H.-T. Hwang, C.-C. Lo, Y.-H. Peng, Y. Tsao, and H.-M. Wang, "Unsupervised representation disentanglement using cross domain features and adversarial learning in variational autoencoder based voice conversion," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 4, pp. 468–479, 2020.

[13] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "Cyclegan-vc3: Examining and improving cyclegan-vcs for mel-spectrogram conversion," *arXiv preprint arXiv:2010.11672*, 2020.

[14] ——, "Stargan-vc2: Rethinking conditional methods for stargan-based voice conversion," *arXiv preprint arXiv:1907.12279*, 2019.

[15] S.-w. Park, D.-y. Kim, and M.-c. Joe, "Cotatron: Transcription-guided speech encoder for any-to-many voice conversion without parallel data," *arXiv preprint arXiv:2005.03295*, 2020.

[16] K. Tanaka, H. Kameoka, T. Kaneko, and N. Hojo, "Atts2s-vc: Sequence-to-sequence voice conversion with attention and context preservation mechanisms," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6805–6809.

[17] W.-C. Huang, T. Hayashi, Y.-C. Wu, H. Kameoka, and T. Toda, "Voice transformer network: Sequence-to-sequence voice conversion using transformer with text-to-speech pretraining," *arXiv preprint arXiv:1912.06813*, 2019.

[18] 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.

[19] 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.

[20] 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.

[21] Y. Y. Lin, C.-M. Chien, J.-H. Lin, H.-y. Lee, and L.-s. Lee, "Fragmentvc: Any-to-any voice conversion by end-to-end extracting and fusing fine-grained voice fragments with attention," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5939–5943.

[22] J. S. Chung, J. Huh, S. Mun, M. Lee, H. S. Heo, S. Choe, C. Ham, S. Jung, B.-J. Lee, and I. Han, "In defence of metric learning for speaker recognition," *arXiv preprint arXiv:2003.11982*, 2020.

[23] J. Kim, J. Kong, and J. Son, "Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech," in *International Conference on Machine Learning*. PMLR, 2021, pp. 5530–5540.

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

[25] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *arXiv preprint arXiv:2010.05646*, 2020.

[26] D. Rezende and S. Mohamed, "Variational inference with normalizing flows," in *International conference on machine learning*. PMLR, 2015, pp. 1530–1538.

[27] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[28] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2794–2802.

[29] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *International conference on machine learning*. PMLR, 2016, pp. 1558–1566.

[30] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brébisson, Y. Bengio, and A. C. Courville, "Melgan: Generative adversarial networks for conditional waveform synthesis," *Advances in neural information processing systems*, vol. 32, 2019.

[31] C. Veaux, J. Yamagishi, K. MacDonald *et al.*, "Superseded-cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit," 2016.

[32] M. Bińkowski, J. Donahue, S. Dieleman, A. Clark, E. Elsen, N. Casagrande, L. C. Cobo, and K. Simonyan, "High fidelity speech synthesis with adversarial networks," *arXiv preprint arXiv:1909.11646*, 2019.

[33] Y. Gu, Z. Zhang, X. Yi, and X. Zhao, "Mediumvc: Any-to-any voice conversion using synthetic specific-speaker speeches as intermedium features," *arXiv preprint arXiv:2110.02500*, 2021.

[34] B. Zhang, D. Wu, C. Yang, X. Chen, Z. Peng, X. Wang, Z. Yao, X. Wang, F. Yu, L. Xie *et al.*, "Wenet: Production first and production ready end-to-end speech recognition toolkit," *arXiv e-prints*, pp. arXiv–2102, 2021.