Interspeech 2022
18-22 September 2022, Incheon, Korea

ACNN-VC: Utilizing Adaptive Convolution Neural Network for One-Shot Voice Conversion

Ji Sub Um, Yeunju Choi, Hoirin Kim

School of Electrical Engineering, KAIST, Daejeon, Republic of Korea
{twiz0311,wkadldppdy,hoirkim}@kaist.ac.kr

Abstract

Voice conversion (VC) converts speaker characteristics of a source speaker to ones of a target speaker without modifying the linguistic content. To overcome limitations of the existing VC systems for target speakers unseen during training, many researchers have recently studied one-shot voice conversion. Although many papers have shown that voice conversion can be performed even with only one utterance of an unseen target speaker, it sounds still far from the target speaker's voice. To enhance the similarity of the generated speech, we implement an adaptive convolution neural network (ACNN) for the voice conversion system in two ways. Firstly, we utilize ACNNs with a normalization method to adapt speaker-related information in denormalization process. The second method is to build an architecture with ACNNs that have various receptive fields to generate a voice closer to the target speaker while considering temporal patterns. We evaluated two methods through objective and subjective evaluation metrics. Results show that the converted speech is better than the previous methods in terms of the speaker similarity while keeping the naturalness score.

Index Terms: voice conversion, one-shot, adaptive convolution neural network

1. Introduction

Voice conversion (VC) is a technique that modifies a source speaker's acoustic information to a target speaker's acoustic information with keeping the source speaker's linguistic information. This technique can be used in various real-world applications such as singing voice generation, privacy protection, and entertainment purpose in the game and movie industry.

The conventional voice conversion systems can be divided into two types according to the condition of used data: parallel VC and non-parallel VC. Parallel VC uses parallel data between source speakers and target speakers. The parallel data are aligned with dynamic time warping algorithm and various approaches such as Gaussian mixture model (GMM) [1, 2], non-negative matrix factorization [3, 4], neural networks [5, 6] have been used to conduct voice conversion with these data. However, parallel data is difficult and expensive to collect. Thus non-parallel VC system appeared to solve the problem. Various approaches such as generative adversarial network (GAN) based VC [7, 8, 9, 10], variational autoencoder (VAE) based VC [11, 12], and automatic speech recognition (ASR) based VC [13] have been proposed. However, they can convert only a single target speaker or limited multi-target speakers and need to be trained again for converting to other target speakers.

To address this issue, one-shot learning scheme started to be used in the VC domain. One-shot VC uses only one target speaker's utterance to extract the speaker embedding at the inference stage. To obtain accurate speaker information using only one utterance, several approaches have been studied to disentangle the speaker information and the content information. [14] uses speaker embedding obtained from global style token (GST) [15] capturing the speaker characteristics globally. [16] proposes a bottleneck network that forces outputs to fill only the content information. [17] tries one-shot VC using adaptive instance normalization technique proposed in the computer vision [18]. [19] uses a vector quantization (VQ) based content encoder trained with contrastive predictive coding (CPC) loss for obtaining linguistic features and mutual information loss to disentangle the content, speaker, and pitch representations.

Although many studies have improved the performance of one-shot VC, the generated voices are still far from the target speaker. We introduce adaptive convolution neural networks (ACNNs) to enhance the similarity by propagating speaker information throughout the network. ACNNs are used to solve several problems in various fields such as denoising [20], video image prediction [21], neural style transfer (NST) [22], and speaker verification [23]. In particular, [22] uses ACNNs to transfer style properties without ignoring local structure. [23] uses ACNNs for capturing phonemes and text-varying characteristics of speech and improving speaker verification performance. Due to the advantage of adapting the desired information with considering local characteristics, we utilize ACNNs with different kernels and biases depending on the speaker embedding to improve the similarity of the converted speech.

In this paper, we apply ACNNs in two ways to the baseline model, VQMIVC [19] which is one of the state-of-the-art models. The main contributions of the paper are as follows:

- We implement ACNNs as a normalization technique to propagate speaker information throughout the network, as in [22].
- We build a new architecture using ACNNs in multi-receptive field fusion modules instead of CNNs. The kernels and biases of ACNNs are decided corresponding to the speaker embeddings.
- Objective and subjective metrics prove the effects of applying ACNNs in terms of the naturalness and the speaker similarity.

2. Preliminary Work

In this section, we introduce the baseline model, VQMIVC. It consists of two encoders, a pitch extractor, a decoder, and variational approximation networks. Each of the two encoders is a content encoder that contains linguistic information and a speaker encoder that extracts speaker-related information. Pitch extractor obtains log-normalized $F0s$ for detecting changes in intonation.

The content encoder receives a mel-spectrogram $X_{src} = \{x_1, x_2, \cdots, x_T\}$ as an input. T denotes the length on the time

10.21437/Interspeech.2022-10473

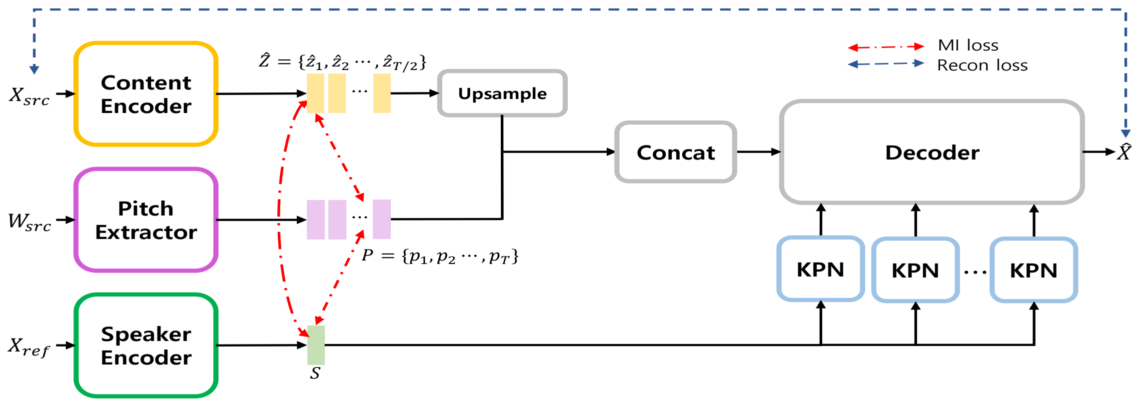

Figure 1: *Overview of our proposed ACNN-VC system. $\hat{Z}$, P, and S denotes the content features, the pitch features, and the speaker embedding, respectively. T is the length on the time axis. X_{src}, W_{src}, and X_{ref} means the source mel-spectrogram, the source waveform, and the reference mel-spectrogram, respectively. KPN denotes a kernel prediction network.*

axis. After passing several layers in the content encoder, this encoder outputs mid-level feature vectors before the vector quantization (VQ) operation. We call it $Z = \{z_1, z_2, \cdots, z_{T/2}\}$. VQ operation converts the inputs to learnable discrete latent variables, which causes unnecessary information to be blown away, leaving only linguistic information. Through this operation, the input Z changes to the closest of the K discrete latent variables, and the output after the VQ operation is called $\hat{Z} = \{\hat{z}_1, \hat{z}_2, \cdots, \hat{z}_{T/2}\}$. To train the learnable discrete latent variables, we use the following VQ loss:

$$L_{VQ} = ||Z - sg(\hat{Z})||_2^2, \tag{1}$$

where sg($\cdot$) means a stop-gradient operator. In addition, this system additionally uses CPC loss to ensure that $\hat{Z}$ contains more accurate linguistic information. To apply CPC loss, a RNN layer is additionally added by using $\hat{Z}$ as input and outputs $C = \{c_1, c_2, \cdots, c_{T/2}\}$. We assume that $\hat{Z}$ values after m steps from time t are positive samples and $\Omega_{t,m}$ is a set of future values except for the value after m steps from time t. The model is trained to distinguish negative samples extracted from the set $\Omega_{t,m}$ and positive samples using outputs of the RNN layer, C. To do this, we use the following CPC loss:

$$L_{CPC} = -\frac{1}{M}\sum_{m=1}^{M} log \frac{exp(\hat{z}_{t+m}^T W_m c_t)}{\sum_{\tilde{z}\in\Omega_{t,m}} exp(\tilde{z}^T W_m c_t)}, \tag{2}$$

where W_m is a trainable linear transformation matrix.

The decoder restores the mel-spectrogram by using the previously obtained $\hat{Z}$ and outputs of the pitch extractor and the speaker encoder. Postnet refines the restored mel-spectrogram. We call the mel-spectrogram before Postnet $\hat{X}$ and the one after Postnet $\tilde{X}$, respectively. The model is trained to reconstruct the true mel-spectrograms using the L2 distance and L1 distance. The reconstruction loss is as follows:

$$\begin{aligned} L_{REC} =& ||\hat{X} - X_{src}||_1 + ||\hat{X} - X_{src}||_2 \\ &+ ||\tilde{X} - X_{src}||_1 + ||\tilde{X} - X_{src}||_2. \end{aligned} \tag{3}$$

Mutual Information (MI) loss is used to prevent the outputs of two encoders and the pitch extractor from sharing the same information. Since MI is difficult to calculate directly, various researches have been proposed to obtain approximation. Among them, vCLUB [24] is used to obtain the upper bound of MI. The upper bound of MI is as follows:

$$\begin{aligned} I(u,v) =& E_{p(u,v)}[logQ_{\theta_{u,v}}(u|v)] \\ &- E_{p(u)}E_{p(v)}[logQ_{\theta_{u,v}}(u|v)], \end{aligned} \tag{4}$$

where $u, v \in \{\hat{Z}, S, P\}$. Each element denotes the outputs of the content encoder, speaker encoder, and pitch extractor. By reducing the upper bound, the encoders are trained to reduce the dependency between outputs. The sum of the unbiased estimations between the outputs using vCLUB estimator is as follows:

$$L_{MI} = \hat{I}(\hat{Z}, S) + \hat{I}(P, S) + \hat{I}(\hat{Z}, P). \tag{5}$$

See equations in [19] for details of MI loss.

To approximate MI by using vCLUB, we need to train three variational approximation networks and the VC system alternatively. To do this, we firstly train variational approximation networks with the following loss:

$$L = logQ_{\theta_{u,v}}(u|v) \tag{6}$$

where $u, v \in \{\hat{Z}, S, P\}$. Whereas, the VC system is trained with the sum of the following losses:

$$L_{total} = L_{REC} + L_{VQ} + L_{CPC} + \lambda_{MI} L_{MI}, \tag{7}$$

where λ_{MI} is a constant weight for MI loss.

3. Methods

Unlike the baseline model in which the outputs of the encoders and the extractor are concatenated, our proposed model utilizes ACNNs in the decoder to make better use of speaker information. Figure 1 shows the overall process of our proposed model. As shown in Figure 1, the speaker embeddings pass through kernel prediction networks (KPNs) instead of concatenation. The kernel prediction network consists of two linear layers. Each linear layer yields kernels and biases of ACNN corresponding to the speaker embedding. Then, convolution is operated by using these parameters. The process is shown in Figure 2. We implement ACNNs to the decoder in two ways.

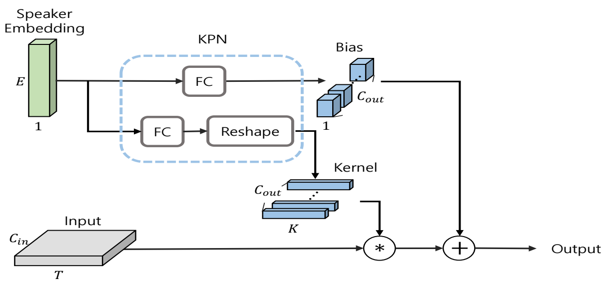

Figure 2: *Architecture of the ACNN. E, K, and T denote the dimension of the speaker embedding, the kernel size, and the time dimension of the input, respectively. C_{in} and C_{out} are the channel size of the input and output. FC is a fully-connected layer. KPN is a kernel prediction network. $*$ means depthwise convolution operation. $+$ operates in a channel-wise manner.*

3.1. Adaptive convolution neural network

Firstly, we use ACNNs with normalization layers. In VC domain, instance normalization (IN) is usually used to remove global statistics in the normalization process and adapt to the desired information in the denormalization process [18, 25]. When the input h passes vanilla IN, the equation is as follows:

$$IN(h) = \alpha\Big(\frac{h-\mu(h)}{\sigma(h)}\Big) + \beta, \tag{8}$$

where $\mu(h)$ means the channel-wise mean vector and $\sigma(h)$ means the channel-wise standard deviation. α and β are scaling factor and shifting factor for the affine transformation. By changing these factors according to the specific features, the model can adapt the desired information. Instead of using affine parameters, [22] introduces 2D style filters, called Adaconv, to use as functions of the denormalization process. Inspired by this normalization method, we implement speaker-related kernels and use them for 1D time features. They are obtained from kernel prediction networks by using speaker embeddings, and the normalized features are convolved with the kernels in the denormalization process. According to the following process, the modified formula is as follows:

$$AdaConv(h) = f * \Big(\frac{h-\mu(h)}{\sigma(h)}\Big) + \beta, \tag{9}$$

where f and β means speaker-related kernels and biases respectively. Also, $*$ denotes depthwise convolution operation. It can transfer speaker-related characteristics to the network considering the local structure on the time axis. We call this model ACNN-VC1 for the rest of the paper.

In the second method, we propose a new architecture with ACNNs. We incorporate ACNNs with a multi-receptive field fusion (MRF) module. This module is firstly introduced in [26]. The MRF module has several standard CNNs with various kernel sizes and dilation sizes, so it has the advantage of identifying patterns of various lengths in parallel. By using ACNNs instead of standard CNNs in the MRF module, the model can capture patterns of various lengths as well as adapt to the target speaker. As a result, it generates sufficiently reasonable mel-spectrograms for the unseen target speaker. Figure 3 shows the block diagram of the MRF module used in our experiments. The kernels and biases of ACNNs in MRF moduels are also decided depending on the speaker embeddings. We call this model ACNN-VC2 for the rest of the paper.

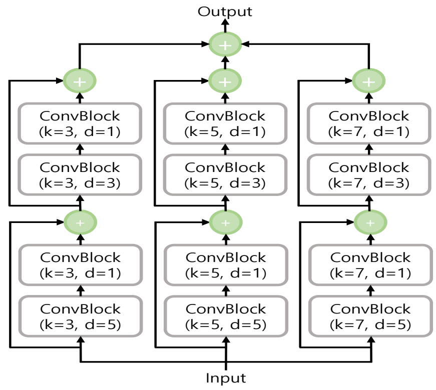

Figure 3: *Architecture of the MRF module. d and k denote the dilation size and the kernel size, respectively. ConvBlock consists of ACNN, batch normalization and a ReLU function.*

3.2. Inference

After the training is completed, we use one utterance of the target speaker as a reference utterance for conversion and pass it into the speaker encoder. Pitch and content vectors are obtained through an utterance of the source speaker. Three outputs of the encoders and the extractor are used as inputs of the decoder to produce the converted mel-spectrograms. Parallel WaveGAN vocoder [27] is used for generating raw waveforms from mel-spectrograms.

4. Experiments

4.1. Experimental setup

Feature Extraction We conduct the experiments with the VCTK corpus [28]. VCTK corpus consists of about 400 utterances per speaker. There are 110 English speakers, and we divided them into 90 speakers and 20 speakers. 20 speakers are used at inference time and the rest are used as the training set. We used 10 percent of the utterances in the training set as the validation set. We extract 80-dimensional log mel-spectrograms from raw speech downsampled to 16000 Hz with hopping sizes of 160, window sizes of 400, and FFT sizes of 400. We extract F0s using World vocoder [29].
Optimization Adam optimizer [30] is used to train our proposed models. The learning rate starts from 10^{-6} and increases to 10^{-3} after a warm-up stage. It becomes half every 100 epochs after 200 epochs. We train for 500 epochs. Variational approximation networks are trained with the Adam optimizer with a learning rate of 0.0003. We set the future prediction step M to 6, the number of negative samples to 10, and λ_{MI} to 0.01.
Architecture The decoder of ACNN-VC1 consists of a 512-dim LSTM layer, a convolution block, two 1024-dim LSTM layers, and an 80-dim linear layer. Postnet is added after the decoder and consists of five convolution layers. Convolution layers in the decoder have a kernel size of 5 and a filter size of 512. We insert the Adaconv layer after every layer except for the last layer in the decoder and Postnet. The decoder of ACNN-VC2 consists of two 512-dim LSTM layers, three MRF modules, an 80-dim linear layer, and the Postnet with five convolution layers. The detail of ACNNs in MRF modules can be confirmed in Figure 3. Postnet consists of 4 convolution layers with a filter size of 512 and a kernel size of 5 and a convolution layer with a filter size of 80 and a kernel size of 5. The

Table 1: *Objective scores of various systems in unseen scenarios: speaker similarity, WER, CER, and predicted MOS.*

Methods	Similarity	WER	CER	pMOS
Ground Truth	0.857	22.81	8.19	4.163
AGAIN-VC	0.694	64.00	32.42	3.161
VQMIVC	0.696	48.69	22.97	3.217
ACNN-VC1	0.714	43.64	19.76	3.455
ACNN-VC2	0.740	42.90	19.21	3.246

encoders, the pitch extractor and the variational approximation networks in our proposed VC systems are the same as [19]. The only difference is that the dimension of the speaker embedding is 128 when training ACNN-VC2. The kernel prediction networks consist of two linear layers.

4.2. Evaluation metric

Subjective and objective evaluations are conducted to compare the proposed methods with several baseline models: AGAIN-VC [25] and VQMIVC [19]. We provide converted samples and ground truth samples on our demo page[1].

We conduct mean opinion score (MOS) tests as subjective evaluations. There are 2 types of MOS: Naturalness and speaker similarity. In naturalness MOS, the converted speech is evaluated on a 5-point scale, from 1 (very bad) to 5 (very good). In speaker similarity MOS, the listeners hear the true target voice and the converted voice and are asked to indicate the degree of similarity between the two voices on a 5-point scale, from 1 (definitely different) to 5 (definitely same). Both metrics are evaluated through Amazon mechanical turk.

We conduct speaker similarity, WER, CER, and predicted MOS (pMOS) scores as objective evaluations. For WER and CER measurements, we use a wav2vec 2.0[2] [31] based automatic speech recognizer (ASR), which is trained with the 960h Librispeech dataset. The pMOS is measured through a neural network based MOS prediction model called NISQA[3] [32]. We measured cosine similarity between the speaker embeddings extracted from the true utterance of the target speaker and converted utterance by using the pre-trained speaker verification model in the open library called Resemblyzer[4] [33]. We use this value as speaker similarity.

4.3. Objective evaluation

In this section, we conduct an objective evaluation for comparing our proposed models to the baseline models. We conduct experiments after dividing 10 out of 20 test speakers into target speakers and 10 into source speakers. We generate a total of 10000 audio samples by constructing 100 pairs. Samples used for a generation are randomly extracted.

As found in Table 1, our proposed methods have lower CER and WER, which means pronouncing words more accurately than baseline models. Also, they have higher pMOS and speaker similarity scores than baseline approaches. It indicates that the converted voices sound more natural and closer to the target speaker. Through these results, we can observe that ACNNs can help the model adapt to the unseen target speaker while keeping the naturalness.

[1]https://twiz0311.github.io/ACNN-VC/
[2]https://huggingface.co/
[3]https://github.com/gabrielmittag/NISQA
[4]https://github.com/resemble-ai/Resemblyzer

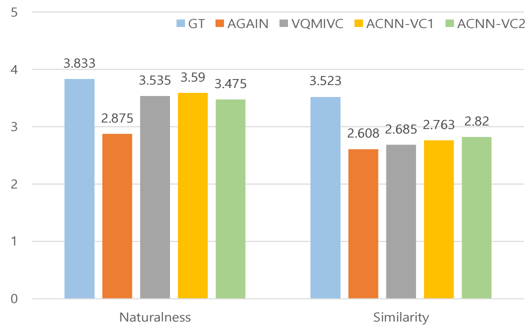

Figure 4: *MOS scores of various systems in unseen scenarios.*

4.4. Subjective evaluation

We conduct naturalness MOS and speaker similarity MOS on 10 raters. We conduct 4 types of voice conversion in unseen domains: female-to-female, female-to-male, male-to-female, and male-to-male. Total 8 speakers were randomly chosen from the test set, and we generate samples by configuring two source-target pairs for each type using 8 speakers. We choose randomly 5 samples for each pair. Thus, a total of 40 samples per system are evaluated.

Figure 4 shows the MOS scores on naturalness and speaker similarity. As shown in this figure, ACNN-VC1 and ACNN-VC2 have higher naturalness MOS than AGAIN-VC and on a par with VQMIVC. Also, both proposed methods have higher speaker similarity MOS than the baseline models. We can observe that these methods can produce voices closer to the target speaker than baseline approaches while keeping the voices still sound natural. That is, applying ACNNs can propagate speaker-related information to the network much better without loss of linguistic information. Also, we can observe that ACNN-VC2 has higher speaker similarity MOS than ACNN-VC1. ACNN-VC1 operates convolution through fixed kernel size and dilation size. On the other hand, ACNN-VC2 conduct convolution operation with various kernel size and dilation size. This allows ACNN-VC2 to apply speaker information to patterns of various lengths and improves speaker similarity better than ACNN-VC1.

5. Conclusions

In this paper, we apply ACNNs to the baseline system in two ways. We use them as functions in the denormalization processes or members of MRFs. When we compare our methods with other VC systems in unseen scenarios, the objective metrics on the converted samples verify our proposed methods can produce more similar sound to the target speaker and have lower WER and CER, which means that they have more accurate pronunciation. Also, the subjective metrics show that the samples converted through our methods sound closer to the target speaker while keeping naturalness. It means that ACNNs help the model adapt the speaker information properly.

6. Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2021R1A2C1014044).

7. 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] R. Takashima, T. Takiguchi, and Y. Ariki, "Exemplar-based voice conversion using sparse representation in noisy environments," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 96, no. 10, pp. 1946–1953, 2013.

[4] Z. Wu, T. Virtanen, E. S. Chng, and H. Li, "Exemplar-based sparse representation with residual compensation for voice conversion," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 10, pp. 1506–1521, 2014.

[5] S. Desai, A. W. Black, B. Yegnanarayana, and K. Prahallad, "Spectral mapping using artificial neural networks for voice conversion," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 5, pp. 954–964, 2010.

[6] L.-H. Chen, Z.-H. Ling, L.-J. Liu, and L.-R. Dai, "Voice conversion using deep neural networks with layer-wise generative training," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1859–1872, 2014.

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

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

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

[10] 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.

[11] 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 *Proc. of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2016, pp. 1–6.

[12] A. Van Den Oord, O. Vinyals *et al.*, "Neural discrete representation learning," in *Proc. of Neural Information Processing Systems (NIPS)*, 2017.

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

[14] H. Lu, Z. Wu, D. Dai, R. Li, S. Kang, J. Jia, and H. Meng, "One-shot voice conversion with global speaker embeddings." in *Proc. of Interspeech*, 2019, pp. 669–673.

[15] Y. Wang, D. Stanton, Y. Zhang, R.-S. Ryan, E. Battenberg, J. Shor, Y. Xiao, Y. Jia, F. Ren, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *Proc. of International Conference on Machine Learning (ICML)*, 2018, pp. 5180–5189.

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

[17] J.-c. Chou and H.-Y. Lee, "One-shot voice conversion by separating speaker and content representations with instance normalization," in *Proc. of Interspeech*, 2019, pp. 664–668.

[18] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1501–1510.

[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," in *Proc. of Interspeech*, 2021, pp. 1344–1348.

[20] B. Mildenhall, J. T. Barron, J. Chen, D. Sharlet, R. Ng, and R. Carroll, "Burst denoising with kernel prediction networks," in *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 2502–2510.

[21] X. Jia, B. De Brabandere, T. Tuytelaars, and L. V. Gool, "Dynamic filter networks," in *Proc. of Neural Information Processing Systems (NIPS)*, 2016.

[22] P. Chandran, G. Zoss, P. Gotardo, M. Gross, and D. Bradley, "Adaptive convolutions for structure-aware style transfer," in *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 7972–7981.

[23] S.-H. Kim and Y.-H. Park, "Adaptive convolutional neural network for text-independent speaker recognition," in *Proc. of Interspeech*, 2021, pp. 641–645.

[24] P. Cheng, W. Hao, S. Dai, J. Liu, Z. Gan, and L. Carin, "Club: A contrastive log-ratio upper bound of mutual information," in *Proc. of International Conference on Machine Learning (ICML)*, 2020, pp. 1779–1788.

[25] Y.-H. Chen, D.-Y. Wu, T.-H. Wu, and H.-y. Lee, "Again-vc: A one-shot voice conversion using activation guidance and adaptive instance normalization," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 5954–5958.

[26] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," in *Proc. of Neural Information Processing Systems (NIPS)*, 2020, pp. 17 022–17 033.

[27] R. Yamamoto, E. Song, and J.-M. Kim, "Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6199–6203.

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

[29] 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.

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[31] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Proc. of Neural Information Processing Systems (NIPS)*, 2020, pp. 12 449–12 460.

[32] G. Mittag, B. Naderi, A. Chehadi, and S. Möller, "Nisqa: A deep cnn-self-attention model for multidimensional speech quality prediction with crowdsourced datasets," *arXiv preprint arXiv:2104.09494*, 2021.

[33] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4879–4883.