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

# Two-Pathway Style Embedding for Arbitrary Voice Conversion

*Xuexin Xu*[1], *Liang Shi*[1], *Jinhui Chen*[2], *Xunquan Chen*[3], *Jie Lian*[1], *Pingyuan Lin*[1], *Zhihong Zhang*[1,*], *Edwin R. Hancock*[4]

[1]School of Informatics, Xiamen University, China
[2]Prefectural University of Hiroshima, Japan
[3]Kobe University, Japan
[4]University of York, United Kingdom

xxxin555@stu.xmu.edu.cn, zhihong@xmu.edu.cn

## Abstract

Arbitrary voice conversion, also referred to as zero-shot voice conversion, has recently attracted increased attention in the literature. Although disentangling the linguistic and style representations for acoustic features is an effective way to achieve zero-shot voice conversion, the problem of how to convert to a natural speaker style is challenging because of the intrinsic variabilities of speech and the difficulties of completely decoupling them. For this reason, in this paper, we propose a Two-Pathway Style Embedding Voice Conversion framework (TPSE-VC) for realistic and natural speech conversion. The novel feature of this method is to simultaneously embed sentence-level and phoneme-level style information. A novel attention mechanism is proposed to implement the implicit alignment for timbre style and phoneme content, further embedding a phoneme-level style representation. In addition, we consider embedding the complete set of time steps of audio style into a fixed-length vector to obtain the sentence-level style representation. Moreover, TPSE-VC does not require any pre-trained models, and is only trained with non-parallel speech data. Experimental results demonstrate that the proposed TPSE-VC outperforms the state-of-the-art results on zero-shot voice conversion.

**Index Terms**: voice conversion, zero-shot learning, attention mechanism, adversarial learning

## 1. Introduction

Voice conversion (VC) is a pervasive task in many areas of speech processing, and aims to convert a certain characteristic of the speech while preserving its linguistic content. The intrinsic variabilities of speech pose different challenges to the VC task, including speaker identity [1], accent [2], emotion [3, 4] and pronunciation [5, 6], *etc*.

Early work on speaker identity conversion was focused on parallel training data, where the speech of the same linguistic content for different speakers is available. Thus, it is possible to learn direct mapping when the acoustic features of the source speech and the target are aligned. However, there was a major problem since the above methods all require parallel data with frame-level alignment, which is both difficult and time-consuming to collect and limits the generalizability of the learned model. Therefore, it is important to develop a method that is capable of adopting the non-parallel training data. Recently, generative models have enjoyed great success, such as variational autoencoders (VAEs) [7] and generative adversarial networks (GANs) [8], and these have performed well on non-parallel training corpus [9, 10, 11, 12, 13]. However, when encountering unseen speakers (not present in the training data), they have demonstrated limited conversion capabilities.

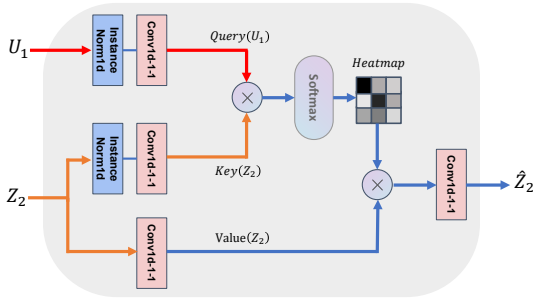



Figure 1: *SPAttention module.*

In zero-shot learning, the learners can handle the classes not previously seen at the training stage. Many approaches have been explored for dealing with unseen data in the VC task. In particular, disentangling the linguistic content information and speaker style information from an input has been proved to be an effective way to achieve zero-shot voice conversion. The proviso here is that the source speaker style representation is replaced by that of the target in the conversion stage. Instance normalization layer [14], carefully designed bottleneck [15], phoneme transcription guidance [16], and vector quantization [17] all implement feature disentanglement in one form or another. However, most of them only embed the speaker style information into a fixed-length vector, which is an average style representation over all time steps. This leads to several problems. For instance, silence segments affect the representation because they contain almost no useful information. On the other hand, synthetic speech has poor naturalness and similarity measurements, which usually accounts for the lack of specific style changes in phonemic content. Moreover, it is often too difficult to decouple the style and content information completely. The residual content information in style may contaminate the representation, so that it limits the quality of the converted speech.

In this paper, we contribute to solving problems encountered with zero-shot voice conversion. To our knowledge, only a few methods (*e.g.* AdaIN-VC [14], AutoVC [15]) can process zero-shot voice conversion, but most of them cannot obtain the availability of both local and global style information. Therefore, the proposed method not only achieves the zero-shot transfer, but also simultaneously models for both local and global style information. For **local** style, we propose a novel attention mechanism, which we refer to as ***"Style to Phoneme Attention (SPAttenton)"***, which both implements an implicit alignment and embeds the most similar speaker style representation for each content phoneme. This fine-grained phoneme-level style

*Corresponding Author



http://dx.doi.org/10.21437/Interspeech.2021-506

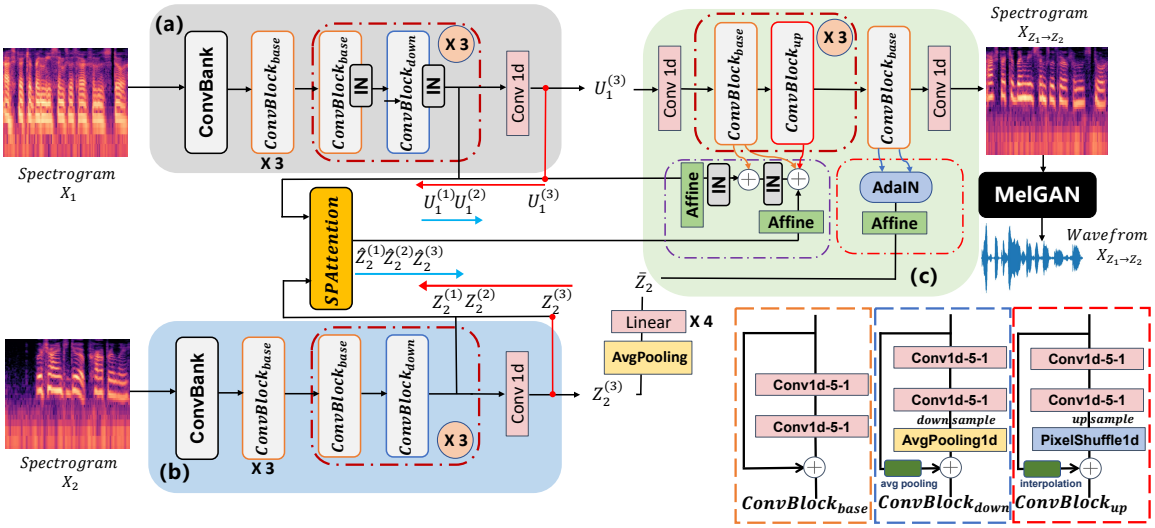



Figure 2: *The generator architecture of TPSE-VC. (a) The content encoder. (b) The style encoder. (c) The decoder. Where $X_1$ and $X_2$ indicate the source and target utterances respectively. According to Eq. (2), the outputs of encoders in different time resolutions will be fed into the SPAttention modules, and $Z_2^{(3)}$ will be averaged to get the sentence-level style representation $\overline{Z}_2$. Finally, in (c), we fuse $U_1$ and $\hat{Z}_2$ to restore the original time steps gradually, then we use the $AdaIN$ operator to embed $\overline{Z}_2$ into the converted speech.*

information mainly encapsulates the target style which depends on source content. For **global** style, we also embed the complete set of time steps of speaker style information into a fixed-length vector, which gives a coarse-grained style information representation. The resulting sentence-level style information encapsulates the style of the whole utterance. To preserve the content information while making significant transformations of the target style, we use a U-Net like [18] multi-scale architecture. We sample or fuse different features at each individual time resolution, and following this, we add sentence-level style information using the AdaIN [19] operator. To further enhance the universality and performance of the method, we only use non-parallel data for training. We encapsulate our voice conversion framework into an adversarial architecture, and use the neural-network-based vocoder MelGAN [20] to construct the transformed speech waveform. Our method does not require any pre-trained models, and generates the speech spectrogram in a non-autoregressive manner. Experimental results indicate TPSE-VC outperforms other SOTA approaches in both subjective and objective evaluations.

## 2. Style to Phoneme Attention

To obtain phoneme-level style information, the key idea is an *attention mechanism* [21] relating style to content [22, 23]. Our approach similarly assumes that the style information is related to the content, so instead of only using a fixed-length vector to represent the style of the whole utterance, the style information should rely on the content and change with time.

As shown in Figure 1, a novel attention module (SPAttention) has been developed to meet the hypothesis above. Accordingly, let $Z_2$ denote the speaker style information of a target utterance and this should depend on the source content representation $U_1$. First, we normalize the input features and transform them linearly, giving *Query*($U_1$), *Key*($Z_2$), and *Value*($Z_2$) denoted by $q$, $K$, and $V$ respectively. Then we use $q$ and $K$ to calculate the attention heatmap to align different phonemic speech content implicitly. Subsequently, we calculate the corresponding speaker style feature $\hat{Z}_2$ which depends on $U_1$ by taking the dot product of $V$ and the attention heatmap. Mathematically, we express this as follows:

$$\hat{Z}_2(t) = \frac{\sum_{t'=1}^{T'} \exp(q^T(t)K(t'))V(t')}{\sum_{t'=1}^{T'} \exp(q^T(t)K(t'))} \cdot W_o \qquad (1)$$

where $q = W_f \cdot IN(U_1)$, $K = W_h \cdot IN(Z_2)$ and $V = W_g \cdot Z_2$, and $IN$ indicates a mean–variance channel-wise normalization to eliminate style information [14]. $\hat{Z}_2(t)$ is the $t^{th}$ time step of $\hat{Z}_2$, and the length of $\hat{Z}_2$ is the same as $U_1$. Let $T'$ denote the length of $Z_2$, then $t'$ is the index that enumerates all time steps of the target style feature. Further, $W_f$, $W_g$, $W_h$, and $W_o$ above indicate the learned weight matrices, which are implemented as Conv1d layer in which both kernel and stride are of unit length. For each time step of $U_1$, this attention mechanism can automatically align the most similar phonemic pronunciation of target speech, and appropriately generate the target style feature which depends on source phonemic content in a learnable manner. Due to our multi-scale architecture, we create different SPAttention modules to obtain the style representations at the corresponding time resolutions.

## 3. Methodology

Our proposed framework is based on GAN [8] which is composed of a generator and a discriminator typically, the generator is an encoder-decoder module in our work. The generator consists of four modules, a content encoder $En_c(\cdot)$ which captures the linguistic content information from the source utterance; a style encoder $En_s(\cdot)$ that produces a speaking style representation from target speech; *SPAttention*$(\cdot,\cdot)$ modules, which are detailed in Section 2 above, it can generate the content-dependent style information; and a decoder $De(\cdot,\cdot,\cdot)$, it takes the content embedding, the phoneme-level style representation and the averaged sentence-level style representation as inputs, and then synthesizes the converted speech by only changing the source speaking style to the target one.

### 3.1. Network architecture

The architecture of the generator is illustrated in Figure 2. Here the generator is made up entirely of convolution layers in order to operate in a non-autoregressive generative manner. For capturing and restoring speech features at different time resolutions, we adopt a multi-scale architecture similar to U-Net [18]. To enlarge the receptive field and capture long-time-scale information, we also employ the ConvBank layer [24] which stacks convolution layers of different kernel sizes. A set of convolution operations is defined as shown in the bottom right of Figure 2. The *ConvBlock*$_{base}$ consists of two Conv1d layers. Then ReLU nonlinear activation is applied after each convolution layer. Both *ConvBlock*$_{down}$ and *ConvBlock*$_{up}$ are based on *ConvBlock*$_{base}$, the differences are that we append an Avg-Pooling layer for downsampling or a PixelShuffle1d layer [25] for upsampling, and the residual connection [26] will be processed to the corresponding time resolution by using average pooling or nearest neighbor interpolation. In addition, we adopt Instance Normalization (IN) after each convolution layer of the content encoder to eliminate speaking style information [14]. The exceptions here are the ConvBank layer and the last output layer. Note we did not design the speaker encoder specifically to eliminate content information. Due to the fact that SPAttention can embed the most phonetically similar style representation for source content, the residual content information in style may be ignored.

In our work, we adopt different time resolutions. The procedure can be written as follows:

$$
\begin{aligned}
&U_1^{(1)}, ..., U_1^{(l)} = En_c(X_1), \quad Z_2^{(1)}, ..., Z_2^{(l)} = En_s(X_2), \\
&\hat{Z}_2^{(1)}, ..., \hat{Z}_2^{(l)} = SPAttention^{(1)}(U_1^{(1)}, Z_2^{(1)}), ..., \\
&\qquad\qquad SPAttention^{(l)}(U_1^{(l)}, Z_2^{(l)}), \\
&\overline{Z}_2 = AvgPooling(Z_2^{(l)}), \\
&X_{Z_1 \to Z_2} = De(U_1^{(1)}, ..., U_1^{(l)}; \hat{Z}_2^{(1)}, ..., \hat{Z}_2^{(l)}; \overline{Z}_2)
\end{aligned}
\tag{2}
$$

where $X_1$ and $X_2$ are the source and target utterances respectively, the superscript $(l)$ denotes different time resolutions. Both the $U_1^{(l)}$ and $\hat{Z}_2^{(l)}$ are fed into the decoder gradually to generate speech spectrogram in a U-Net like manner. To fuse the global style information, we first use average pooling for different length utterances to obtain fixed-length representations, then feed it into several linear transformations. Later, the AadIN operator is used to embed the sentence-level style.

Unlike the generator, the discriminator is constructed with 2d convolution layers like [27, 28] to better capture the acoustic texture. There are 5 convolution layers with stride 2 and kernel size 5 to downsample the feature map gradually. Then an output layer is appended to match the target channel. Instance Normalization [29] and Leaky ReLU [30] are applied after each convolution layer except the final output layer.

### 3.2. Loss functions

As discussed above, we train this model using only non-parallel data in an unsupervised manner. Thus, we adopt the L1 reconstruction loss between the predicted and ground-true spectrograms when the inputs are the same. We calculate the L2 content loss between the source content features and the converted ones. And the corresponding style distances are computed using the target style features and the converted ones. In adversarial training, the WGAN-GP loss [31] is adopted to mitigate against the training instability issue. The weight of reconstruction loss is fixed at 1, all the rest rely on different weighting parameters denoted by $\lambda_1$, $\lambda_2$ and $\lambda_3$ respectively. The mathematical formulations can be found on `https://github.com/XXxin1/tpse-vc/blob/main/Loss_Function.pdf`

### 3.3. Training details

As shown in Figure 2, we employed a pre-trained MelGAN vocoder [20] to implement the transformation from speech spectrogram to speech waveform. For this, we process the original speech signals in the required format of the MelGAN input. We convert the sample rate of audio into 22,050Hz and perform the short-time Fourier transform (STFT) with 1024 STFT window size, then we transform the magnitude of the spectrograms to 80-bin mel-scale and take logarithm.

We trained the proposed TPSE-VC by ADAM optimizer with 0.0001 as learning rate, and $\beta_1 = 0.9$, $\beta_2 = 0.999$. We set the weight decay to 0.0001 to prevent the model from overfitting. The weighting parameters are simply set as $\lambda_1 = 1$, $\lambda_2 = 1$ and $\lambda_3 = 0.01$ in our experiment. We trained the model for 200k iterations (mini-batch = 128). Further details can be found on: `https://github.com/XXxin1/tpse-vc`

## 4. Experiments

### 4.1. Experiment conditions

We conduct experimental evaluations on the CSTR VCTK Corpus [32], which contains about 44 hours of utterances produced by 109 English speakers with different accents. For the zero-shot non-parallel conversion setting, we randomly selected 20 speakers as the testing set, where they were denoted as unseen speakers. And the rest 89 speakers' utterances will be used to train our proposed model. We first trimmed the audio to reduce the training difficulty, then transformed it into the corresponding acoustic feature. Later, we randomly cropped the acoustic features with the segment length of 128 during training, the variable-length inputs can be processed in the inference stage due to our fully-convolution architecture.

Three baselines, the state-of-the-art arbitrary voice conversion approaches, were adopted for the performance comparison named AutoVC [15], AdaIN-VC [14] and VQVC+ [33]. We reproduced these methods by their open source implementations or the official pre-trained models also trained o VCTK corpus.

To fair comparison with the SOTA methods, the testing pairs were guaranteed that they all were spoken by **unseen** speakers. The four conversion scenarios were average considered (intra/inter-gender), we randomly sampled 2000 testing pairs. Note these pairs have different transcriptions.

Table 1: *Subjective evaluation: The MOS results.*

| **MOS** | (a) **Proposed** | (b) AutoVC | (c) AdaIN-VC | (d) VQVC+ |
|---|---|---|---|---|
| **Nat.** | **3.24** $\pm$ 0.06 | 2.57 $\pm$ 0.06 | 2.65 $\pm$ 0.06 | 2.76 $\pm$ 0.07 |
| **Sim.** | **3.36** $\pm$ 0.06 | 2.26 $\pm$ 0.05 | 2.55 $\pm$ 0.06 | 2.70 $\pm$ 0.06 |

### 4.2. Subjective evaluation

The converted speech was evaluated subjectively in both naturalness and similarity. We randomly and averagely selected 80 testing pairs in the previous pairs for each scenario, and we ensured each scenario with at least 8 subjects. Then we converted these pairs using different methods and ordered them randomly.

Table 2: *Objective evaluation: The speaker (cosine) similarity ($10^{-2}$) generated by a third-party speaker verification system.*

| **Similarity** | Comparison with SOTAs | | | | Ablation Studies | | | Lower and Upper Bounds | |
|---|---|---|---|---|---|---|---|---|---|
| | (a) **Proposed** | (b) AutoVC | (c) AdaIN-VC | (d) VQVC+ | (e) *-adv* | (f) *-sentence* | (g) *-phoneme* | (h) ST | (i) vocoder |
| **Average** ($10^{-2}$) | **77.46** | 62.88 | 76.54 | 68.06 | 64.99 | 74.26 | 66.86 | 55.87 | 93.48 |

* ST & vocoder: the similarity between target utterances and source ones, and between target utterances and the vocoder-reconstructed ones.

We conducted the Mean Opinion Score (MOS) test to evaluate the perceptual quality for speech naturalness and speaker similarity. In the naturalness test, each converted utterance was presented to the listeners who were asked to give a 5-scale opinion score from 1 to 5 how natural the speech sounded (5 is the best). And in the similarity test, each listener was asked to listen to the converted speech and the corresponding target speech, and marked 1 to 5 regarding how confident they thought these two utterances were said by the same speaker. Note such subjective evaluation was conducted on **unseen** speakers, which is considered more important in the real world.

The scores are presented with the 95% confidence intervals in Table 1. Our proposed method achieved the best performance among other approaches in human perceptual evaluations. This result demonstrated that TPSE-VC can synthesize more natural and realistic speech signals than other baselines because of the multi-scale feature fusion, the adversarial training strategy, and the two-pathway style embedding.

### 4.3. Objective evaluation

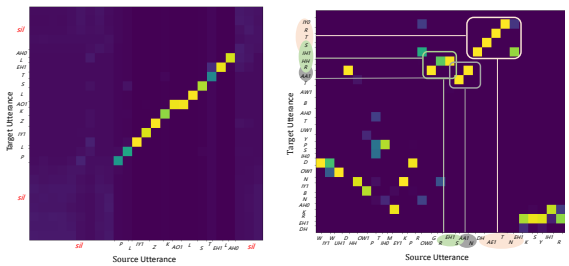
In objective evaluations, an extra speaker verification system Resemblyzer[1] was employed to embed the speaker characteristics into a fixed-length feature. For each sampled pair (2000 pairs total mentioned above), we fed both the target speech and the converted speech into Resemblyzer to get speaker representations. Then we calculated the cosine similarity between them to measure the speaker similarity. Meanwhile, the similarity of the target vocoder-reconstructed utterances and source utterances were also evaluated to become the upper and lower bound for the speaker similarity respectively.

The results are reported in Table 2 (a) $\sim$ (d). As we can see, among the SOTA any-to-any voice conversion methods, TPSE-VC (our proposed) achieved better or comparable performances. The similarity of AdaIN-VC [14] is pretty much the same as ours, but in human perception (*i.e.* subjective evaluations or demo page[2]), the converted utterances generated by our proposed method are much more similar to the target speech.

### 4.4. Ablation studies

In this section, the ablation studies were conducted by removing the different roles in turn. For investigating the effects of the adversarial training, we canceled this strategy by removing the discriminator and the adversarial loss, then we trained the whole generator like a vanilla autoencoder. To indicate the advantages of our phoneme-level and sentence-level style embeddings, we separate these two different style modes. We only used one of them to embed style information for the whole framework by removing the corresponding modules and connections. The different strategies were denoted by *-adv*, *-sentence* (*i.e.* only phoneme-level), and *-phoneme* (*i.e.* only sentence-level) respectively. Table 2 (e) $\sim$ (g) show the objective evaluations of ablation studies by removing different roles, which can confirm the effectiveness of different roles to some extent.

[1] https://github.com/resemble-ai/Resemblyzer
[2] https://xxxin1.github.io/TPSEVC-Demo/




(a) Same content pair (b) Different content pair

Figure 3: *Attention heatmap visualization. For each source phoneme, the SPAttention tries to generate the most phonetically similar style by referring to the target utterance.*

### 4.5. Attention analysis

To present the performances of SPAttention, we analyzed and visualized the heatmap after the softmax activation function to gain meaningful insights. The last attention module was chosen, and we considered two different scenarios: the different speakers with the same and different content.

VCTK corpus has a small amount of parallel data (rainbow passage and the elicitation paragraph). As shown in Figure 3 (a), this visualization was produced by the utterances that were spoken by different speakers with the same content. In the silence (*sil*) parts, we can see the heatmap is averaged, the SPAttention cannot obtain any style information because there were no phonemes that can be recognized. Besides the silence part, diagonal attention can be easily observed due to the same content. In Figure 3 (b), we selected another testing pair with different content from different speakers. It can be found that the phonetically similar corresponding positions are lit, the SPAttention tends to find the phonetically similar style representation for each content phoneme (*e.g.* /EH1/ and /IH1/, /AAI N/ and /AAI R/, and /T N EH1/ and /T R IY0/ & /IH1/) and then constructs the phoneme-level style information.

## 5. Conclusions

In this paper, we have proposed TPSE-VC, a zero-shot non-parallel voice conversion framework, by embedding the fine-grained and coarse-grained style information simultaneously. The proposed TPSE-VC achieved comparable or even better performance than other arbitrary voice conversion methods in both subjective and objective evaluations. In addition, such an attention-based VC framework does not need to ensure the style and content embedding are completely independent of each other, and it can easily achieve zero-shot voice conversion.

# 6. References


[1] A. Kain and M. W. Macon, "Spectral voice conversion for text-to-speech synthesis," in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181)*, vol. 1. IEEE, 1998, pp. 285–288.

[2] D. Felps, H. Bortfeld, and R. Gutierrez-Osuna, "Foreign accent conversion in computer assisted pronunciation training," *Speech communication*, vol. 51, no. 10, pp. 920–932, 2009.

[3] H. Kawanami, Y. Iwami, T. Toda, H. Saruwatari, and K. Shikano, "Gmm-based voice conversion applied to emotional speech synthesis," in *Eighth European Conference on Speech Communication and Technology*, 2003.

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

[5] 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.

[6] L.-W. Chen, H.-Y. Lee, and Y. Tsao, "Generative adversarial networks for unpaired voice transformation on impaired speech," *Proc. Interspeech 2019*, pp. 719–723, 2019.

[7] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, pp. 2672–2680, 2014.

[9] C.-C. Hsu, H.-T. Hwang, Y.-C. Wu, Y. Tsao, and H.-M. Wang, "Voice conversion from unaligned corpora using variational autoencoding wasserstein generative adversarial networks," *Proc. Interspeech 2017*, pp. 3164–3168, 2017.

[10] P. L. Tobing, Y.-C. Wu, T. Hayashi, K. Kobayashi, and T. Toda, "Non-parallel voice conversion with cyclic variational autoencoder," *Proc. Interspeech 2019*, pp. 674–678, 2019.

[11] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "Stargan-vc2: Rethinking conditional methods for stargan-based voice conversion," *Proc. Interspeech 2019*, pp. 679–683, 2019.

[12] Y. Saito, Y. Ijima, K. Nishida, and S. Takamichi, "Non-parallel voice conversion using variational autoencoders conditioned by phonetic posteriorgrams and d-vectors," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5274–5278.

[13] 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.

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

[15] 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.

[16] J.-X. Zhang, Z.-H. Ling, and L.-R. Dai, "Non-parallel sequence-to-sequence voice conversion with disentangled linguistic and speaker representations," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 540–552, 2019.

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

[18] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[19] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1501–1510.

[20] 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, pp. 14 910–14 921, 2019.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[22] D. Y. Park and K. H. Lee, "Arbitrary style transfer with style-attentional networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5880–5888.

[23] T. Ishihara and D. Saito, "Attention-based speaker embeddings for one-shot voice conversion," *Proc. Interspeech 2020*, pp. 806–810, 2020.

[24] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio *et al.*, "Tacotron: Towards end-to-end speech synthesis," *arXiv preprint arXiv:1703.10135*, 2017.

[25] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1874–1883.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[27] J.-c. Chou, C.-c. Yeh, H.-y. Lee, and L.-s. Lee, "Multi-target voice conversion without parallel data by adversarially learning disentangled audio representations," *Proc. Interspeech 2018*, pp. 501–505, 2018.

[28] 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.

[29] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016.

[30] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *International Conference on Machine Learning*, vol. 30, 2013, p. 1.

[31] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in neural information processing systems*, 2017, pp. 5767–5777.

[32] C. Veaux, J. Yamagishi, and K. Macdonald, "Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit," 2017.

[33] D.-Y. Wu, Y.-H. Chen, and H.-y. Lee, "Vqvc+: One-shot voice conversion by vector quantization and u-net architecture," *Proc. Interspeech 2020*, pp. 4691–4695, 2020.