TOWARDS LOW-RESOURCE STARGAN VOICE CONVERSION USING WEIGHT ADAPTIVE INSTANCE NORMALIZATION

Mingjie Chen, Yanpei Shi, Thomas Hain

Department of Computer Science, University of Sheffield,
mchen33, yshi30, t.hain@sheffield.ac.uk

ABSTRACT

Many-to-many voice conversion with non-parallel training data has seen significant progress in recent years. StarGAN-based models have been interests of voice conversion. However, most of the StarGAN-based methods only focused on voice conversion experiments for the situations where the number of speakers was small, and the amount of training data was large. In this work, we aim at improving the data efficiency of the model and achieving a many-to-many non-parallel StarGAN-based voice conversion for a relatively large number of speakers with limited training samples. In order to improve data efficiency, the proposed model uses a speaker encoder for extracting speaker embeddings and conducts adaptive instance normalization (AdaIN) on convolutional weights. Experiments are conducted with 109 speakers under two low-resource situations, where the number of training samples is 20 and 5 per speaker. An objective evaluation shows the proposed model is better than the baseline methods. Furthermore, a subjective evaluation shows that, for both naturalness and similarity, the proposed model outperforms the baseline method.

***Index Terms*—** Voice Conversion, Generative Adversarial Networks, Low-resource

1. INTRODUCTION

Given one voice sample of a source speaker then one voice sample of a target speaker, voice conversion aims at generating one voice sample that contains the speech content information from the source sample while the target speaker's properties. Statistical models such as Gaussian mixture models (GMMs) [1, 2] have been used for voice conversion. Besides, deep neural networks (DNN) [3, 4] have also been popular for voice conversion. However, both the GMM-based models and the DNN-based models required aligned parallel data for training, where source sample and target sample contain the same speech content information. Obtaining aligned parallel data is not easy and requires time-consuming human works. More recently, generative models such as variational auto-encoder [5, 6, 7] (VAE) and generative adversarial network (GAN) [8] have gained attentions for non-parallel voice conversion.

In terms of GAN-based models for non-parallel voice conversion, CycleGAN-VC [9] used CycleGAN [10] model. A cycle-consistency loss was used in CycleGAN-VC to avoid using aligned parallel data. StarGAN-VC [11] proposed to use StarGAN [12] model for voice conversion. It used a domain classifier module, in order to enhance the similarity of converted samples. StarGAN-VC suffered from a partial conversion issue, which means the converted voices were neutral. Also the domain classifier module influenced the voice quality. StarGAN-VC2 [13] and [14] were proposed to improve the performance of StarGAN-based voice conversion by removing the domain classifier module. StarGAN-VC2 proposed to use conditional instance normalization [15] to improve the speaker adaptation ability of the model. However, feature-based normalization layers [16, 15, 17] have been found causing information loss[18], which could lead to low data efficiency.

Most of the mentioned StarGAN-based voice conversion research have used a relatively small number of speakers. For example, in StarGAN-VC and StarGAN-VC2, only 4 speakers were used, and the amount of the training data per speaker was 5 minutes. [19] trained the StarGAN-VC model with 37 speakers, however the training data per speaker was 30 minutes in average. It is unclear whether the StarGAN-based models can keep the performance when increasing the number of speakers and decreasing the training samples.

This work aims at improving the data efficiency of the StarGAN-based model and exploring voice conversion under low-resource situations. We propose a weight adaptive instance normalization StarGAN-VC (WAStarGAN-VC) model. Two approaches are used to improve the data efficiency of the model: (1) unlike StarGAN-VC and StarGAN-VC2 only using speaker identity for target speaker information, we uses a speaker encoder to extract speaker embeddings from target speech; (2) instead of normalizing feature, we follow the idea from StyleGAN2 [18] and conduct adaptive instance normalization on the convolutional weights, to avoid information loss caused by normalization layers. The voice conversion experiments are conducted with 109 speakers under two low-resource situations. We use speaker identification and verification for objective evaluation. For subjective evaluation, we evaluate the proposed model using ABX test (similarity) and AB test (naturalness). The evaluation results show that WAStarGAN-VC outperforms the baseline models.

2. STARGAN-BASED VOICE CONVERSION

This section reviews two previous StarGAN-based voice conversion models: StarGAN-VC [11] model and StarGAN-VC2 model [13].

2.1. StarGAN-VC Model

StarGAN-VC [11] adapted and used the StarGAN [12] model for voice conversion. The model is composed of three modules: a generator $G()$, a discriminator $D()$ and a domain classifier $C()$. Given a real data $x \sim p(x)$ ($p(x)$ is the real data distribution) and a target speaker identity s_y, the generator converts data x to data y.

$$y = G(x, s_y) \tag{1}$$

As shown in Equation 2, the discriminator takes in a data x^* and a speaker identity s^*, where (x^*, s^*) can be real source data and source speaker identity (x, s_x) or converted data and target speaker identity (y, s_y).

$$o = D(x^*, s^*), \tag{2}$$

where o is the output of the discriminator, s_x is the source speaker id. o is the probability that the input x^* belongs to real data distribution.

The loss function of StarGAN-VC is composed of four parts:

$$\mathcal{L}^G_{StarGAN-VC} = \mathcal{L}^G_{adv} + \mathcal{L}^G_{cyc} + \mathcal{L}^G_{id} + \mathcal{L}^G_{domain} \tag{3}$$

$$\mathcal{L}^D_{StarGAN-VC} = \mathcal{L}^D_{adv} \tag{4}$$

The adversarial loss is defined as:

$$\mathcal{L}^G_{adv} = -\mathrm{E}_{x,s_y}[D(G(x, s_y), s_y))] \tag{5}$$

$$\mathcal{L}^D_{adv} = -\mathrm{E}_{x,s_x}[D(x, s_x)] - \mathrm{E}_{x,s_y}[1 - D(G(x, s_y), s_y)] \tag{6}$$

Besides, StarGAN-VC also used the identity loss $\mathcal{L}_{id}$ and the cycle consistency loss $\mathcal{L}_{cyc}$.

$$\mathcal{L}^G_{id} = \mathrm{E}_{x,s_x}[||x - G(x, s_x)||_1] \tag{7}$$

$$\mathcal{L}^G_{cyc} = \mathrm{E}_{x,s_y,s_x}[||x - G(G(x, s_y), s_x)||_1] \tag{8}$$

The domain classifier is used to force the generated data y to be similar to the target speaker s_y.

$$\mathcal{L}^C_{domain} = -\mathrm{E}_{x,s_x}[p_C(s_x|x)] \tag{9}$$

$$\mathcal{L}^G_{domain} = -\mathrm{E}_{x,s_y}[p_C(s_y|G(x, s_y))] \tag{10}$$

2.2. StarGAN-VC2 Model

One of limitations of the StarGAN-VC model is that the domain classifier loss hurts the voice quality [13]. Additionally, only using the target speaker identity s_y in the generator and the discriminator causes the partial conversion issue [13]. In order to solve the voice quality issue, the StarGAN-VC2 model removed the domain classifier module. Besides, to improve similarity, the StarGAN-VC2 model used the concatenation of the source speaker embedding e_x and the target speaker embedding e_y as the speaker condition input to the generator and the discriminator.

$$e_{xy} = concat([e_x, e_y]) \tag{11}$$

where $concat$ is the concatenation function, speaker embeddings e_x an e_y can be obtained through speaker ids s_x and s_y.

StarGAN-VC2 incorporated conditional instance normalization [15] (CIN) in the generator. In the StarGAN-VC2 model, CIN normalizes the feature f across time and conducts affine transformation given the speaker condition e_{xy}.

$$CIN(f) = \gamma(e_{xy}) * (\frac{f - \mu}{\sigma}) + \beta(e_{xy}), \tag{12}$$

where $CIN(f)$ is the output of CIN, $\gamma()$ and $\beta()$ are linear functions, μ and σ are the mean and the standard deviation of the feature f over time.

The training objective of StarGAN-VC2 is similar to StarGAN-VC, including the adversarial loss, the identity loss and the cycle consistency loss. StarGAN-VC2 did not use the domain classifier loss.

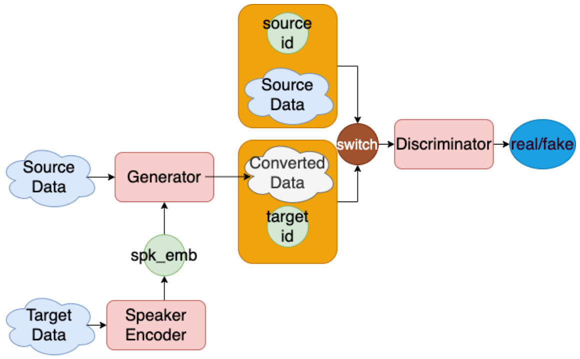

Fig. 1. Model architecture of the proposed WAStarGAN-VC model, *spk_emb* denotes speaker embedding

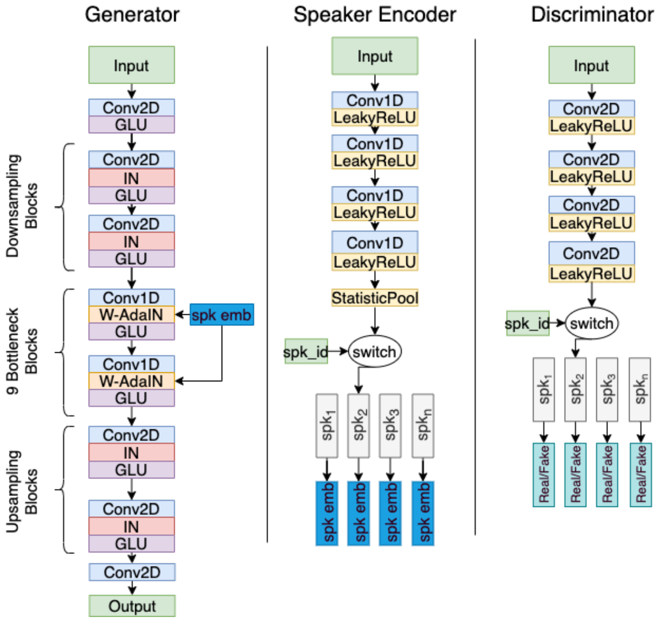

Fig. 2. Module details of the proposed WAStarGAN-VC: spk_id denotes speaker identity, spk_emb denotes speaker embedding

3. STARGAN VOICE CONVERSION WITH WEIGHT ADAPTIVE INSTANCE NORMALIZATION

Given a source data $x_s \sim p(x)$ and a target data $x_t \sim p(x)$, the proposed WAStarGAN-VC model is expected to generate a data y_t that contains the speech content information of x_s and the speaker properties of x_t. WAStarGAN-VC is composed of three modules: a generator $G()$, a discriminator $D()$ and a speaker encoder $E()$.

Both StarGAN-VC and StarGAN-VC2 used speaker identity as the target speaker information input. In contrast, in order to improve the data efficiency of the model, WAStarGAN-VC uses a speaker encoder to extract speaker embeddings from target data. By doing this, the model is expected to learn speaker embeddings more efficiently. On the other hand, it has been found that normalization layers such as instance normalization [16] could cause information loss [18]. WAStarGAN-VC proposes to normalize and transform convolutional weights, to improve the data efficiency of the model as in StyleGAN2[18].

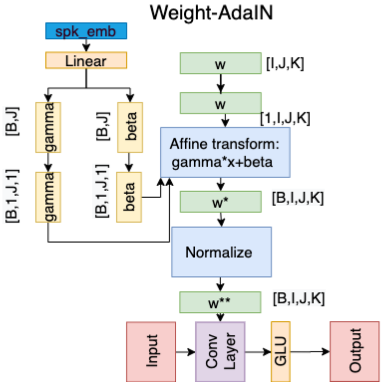

Fig. 3. Weight adaptive instance normalization: *spk_emb* denotes speaker embedding, gamma and beta are affine parameters. 'GLU' denotes the activation function. B, I, J, K are batch size, outcoming channels, incoming channels and kernel size respectively.

3.1. Generator with Weight Adaptive Instance Normalization

WAStarGAN-VC uses a 2-1-2 model architecture for the generator, which is similar to CycleGAN-VC [20] and StarGAN-VC2 [13]. The generator contains three parts: the downsampling blocks, the bottleneck blocks and the upsampling blocks. As shown in Figure 2, the upsampling blocks and the downsampling blocks uses 2D-convolutional layers and instance normalization [16] (IN). There are 9 bottleneck blocks, where each contains a 1D-convolutional layer with the weight adaptive instance normalization (W-AdaIN). The gated linear units (GLU) are used as the activation function.

3.1.1. *Adaptive Instance Normalization*

Adaptive instance normalization [17] (AdaIN) was initially proposed for image style transfer tasks. Based on CIN (Equation 12), AdaIN uses a speaker encoder to extract the speaker embedding $e_y = E(y)$.

$$AdaIN(f) = \gamma(e_y) * (\frac{f-\mu}{\sigma}) + \beta(e_y), \tag{13}$$

where $AdaIN(f)$ is the output of AdaIN, e_y is speaker embedding, y is target data, f is feature, μ and σ are the mean and the standard deviation of the feature f across time, $\gamma()$ and $\beta()$ are linear functions.

3.1.2. *Weight Adaptive Instance Normalization*

This work tries to improve the data efficiency of the model by using the W-AdaIN module in the bottleneck blocks of the generator. In WAStarGAN-VC, as shown in Figure 3, the 1D-convolutional weight w has the shape of $[I, J, K]$, where I is the outcoming channel dimensionality of the convolutional layer, J is the incoming channel dimensionality of the convolutional layer, K is the kernel size.

The target speaker data x_t is fed into the speaker encoder to get the speaker embedding $e_t = E(x_t)$. e_t is fed into linear functions to get the affine parameters γ and β. The affine parameters γ and β have the shape of $[B, J]$, where B is the batch size. Then they are expanded on the second and the fourth dimension.

$$\gamma_{b,1,j,1}, \beta_{b,1,j,1} = \gamma_{b,j}, \beta_{b,j}$$

Then the weight w is expanded on the first dimension, where $w_{i,j,k}$ is the element of w.

$$w_{1,i,j,k} = w_{i,j,k}$$

Next, the expanded weight $w_{1,i,j,k}$ is transformed by $\gamma_{b,1,j,1}$ and $\beta_{b,1,j,1}$.

$$w^*_{b,i,j,k} = \gamma_{b,1,j,1} * w_{1,i,j,k} + \beta_{b,1,j,1} \tag{14}$$

The transformed weights $w^*_{b,i,j,k}$ are normalized across the outcoming dimension (I).

$$w^{**}_{b,i,j,k} = \frac{w^*_{b,i,j,k} - \mu_{b,1,j,k}}{\sigma_{b,1,j,k}}, \tag{15}$$

where $w^{**}_{b,i,j,k}$ is the output of the W-AdaIN module, $\mu_{b,1,j,k}$ and $\sigma_{b,1,j,k}$ are the statistics of $w^*_{b,i,j,k}$ across the outcoming dimension I. Finally, the convolution is conducted on feature using the new adapted weight w^{**}.

3.2. Discriminator and Speaker Encoder

To get speaker-conditioned discriminator output, as in [14] and StarGAN-V2 [21], the discriminator uses N parallel speaker-conditioned output layers, where N is the number of the speakers in the training dataset. As shown in Figure 2, in the discriminator, the first 4 layers are shared across N speakers. For one input sample, the switch selects one of the speaker-conditioned output layers according to the input speaker id. Hence the output of the discriminator is conditioned on the speaker identity. The speaker encoder also uses the speaker-conditioned parallel output layers. Moreover, the speaker encoder uses a statistic pooling layer as in the Xvector [22].

3.3. Training Objectives

In WAStarGAN-VC, the training objectives include three parts: the adversarial loss, the cycle consistency loss and the speaker embedding reconstruction loss. As for the adversarial loss, the least square loss [23] is used, which is the same as in StarGAN-VC2 [13]. The cycle consistency loss is the same as in Equation 8. The speaker embedding reconstruction loss $\mathcal{L}_{spk}$ tries to reconstruct the target speaker embedding e_t from the converted data y_t.

$$\mathcal{L}_{spk} = \mathrm{E}_{x_s,x_t}[||E(x_t) - E(G(x_s, E(x_t)))||_1] \tag{16}$$

4. EXPERIMENT IMPLEMENTATION

The experiments use VCTK [24] dataset [1]. The VCTK dataset contains English speech studio recordings with 109 speakers. The average number of speech samples per speaker is 400.

4.1. Experiment Setup

The experiments are split into three situations according to the number of speakers, the number of training samples: (1) for the first situation, 10 speakers with the full training samples are used, (2) for the second situation, 109 speakers with 20 samples per speaker are used, (3) for the third situation, 109 speakers with 5 samples per speaker are used. StarGAN-VC and StarGAN-VC2 are used as baseline methods. In case that there are no official open source implementations of the StarGAN-VC model and the StarGAN-VC2 model, we implemented two baseline models. The waveform data is downsampled into 22.05 kHz. Mel-cepstral coefficients (MCEPs)

[1] Source code is available at: https://github.com/MingjieChen/LowResourceVC, voice samples is available at: https://minidemo.dcs.shef.ac.uk/wastarganvc/

	N=10,M=Full S=900		N=109,M=20, S=5400		N=109,M=5, S=5400	
Model	ACC	EER	ACC	EER	ACC	EER
StarGAN-VC	64.4	14.88	54.4	21.96	none	none
StarGAN-VC2	91.5	2.99	79.6	4.61	62.6	8.27
Ours	**97.0**	**0.66**	**95.9**	**1.77**	**92.5**	**3.56**

Table 1. Objective evaluation results: ACC (%) denotes speaker identification accuracy, EER (%) denotes the speaker verification equal error rate. N is the number of speakers, M is the number training samples, S is the number of converted samples for evaluation.

are extracted using PyWorld [25] toolkit. The StarGAN-based models only focus on the conversion of the MCEPs. As in [11] and [13], the logarithmic fundamental frequencies (F0s) are transformed linearly. WORLD [25] vocoder is used to generate waveform based on the converted MCEPs, the transformed F0s and the aperiodicities (APs). Finally, the loudness of the generated waveform is normalized using PyLoudNorm [26] toolkit.

4.2. Model Configurations

The proposed WAStarGAN-VC model is implemented using the PyTorch [27] toolkit. The optimizer is Adam [28] with the learning rate for the generator and the discriminator as 2e-4 and 1e-4 respectively. The MCEPs are randomly cropped into 256-frame segments during training. The batch size is 8 and the training process takes 250k iterations for 30 hours on one single GPU.

5. EXPERIMENT RESULTS

The evaluation includes objective evaluation and subjective evaluation. For objective evaluation, we evaluate the models on all three situations. For subjective evaluation, we only evaluate the StarGAN-VC2 model and the WAStarGAN-VC model on the second situation.

5.1. Objective Evaluation

For objective evaluation, as in [29], speaker identification accuracy (ACC) and speaker verification equal error rate (EER) are the measurements of the quality of the converted samples. In this work, a Xvector [22] model is pretrained on the VCTK dataset for the whole 109 speakers. The ACC and EER of the converted samples are used as evaluation metrics. For the third situation where the number of the training samples is 5, the StarGAN-VC model failed to generate sensible voices.

As shown in Table 1, generally, in all three situations, the proposed model yields the best ACC and EER results. For the first situation, WAStarGAN-VC gets ACC 97.0%, EER 0.66%. StarGAN-VC2 is slightly worse than WAStarGAN-VC (ACC 91.5%, EER 2.99%), StarGAN-VC is much worse (ACC 64.4%, EER 14.88%). For the second situation, WAStarGAN-VC gets ACC 95.9%, EER 1.77%. StarGAN-VC2 gets ACC 79.6%, EER 4.61%, and StarGAN-VC gets 54.4%, EER 21.96%. Both two baseline models are much worse for this situation. For the third situation, WAStarGAN-VC gets ACC 92.5%, EER 3.56%. StarGAN-VC2 gets ACC 62.6%, EER 8.27%, which is much worse than the proposed model.

The objective results show that our proposed model is slightly better than StarGAN-VC2 when using the full of training samples for 10 speakers. However, for the low-resource situations, our proposed model is much better than StarGAN-VC and StarGAN-VC2. This maybe because the proposed model has better data efficiency, which enables it being able to keep the performance under the low-resource situations.

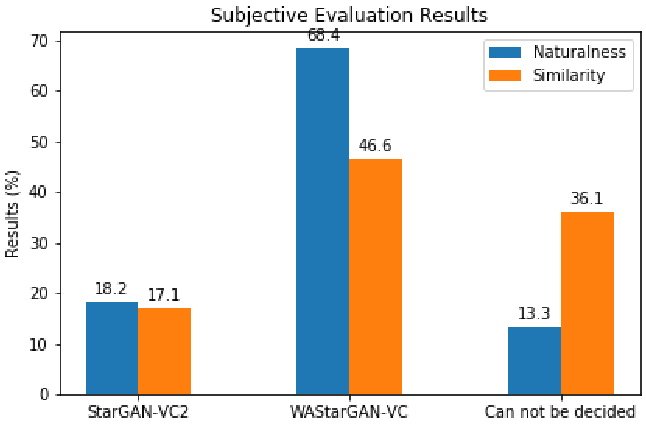

Fig. 4. Subjective evaluation results

5.2. Subjective Evaluation

To assess the naturalness and the similarity, this work conducts the listening tests by comparing WAStarGAN-VC and StarGAN-VC2. The two models are trained under the second situation where the number of speakers is 109 and the number of training samples is 20. AB tests are used for the naturalness evaluation, where evaluators need to choose one sample that has better naturalness from two samples generated from two models. For the similarity evaluation, ABX tests are used. Evaluators need to choose one from two samples that is more similar to the real target sample. A subset of 10 speakers is randomly selected (5 male and 5 female). In total 90 (10*9=90 all conversion directions) samples are evaluated for each model.

As shown in Table 4, for both the naturalness and the similarity, the proposed model obtains the most choices. WAStarGAN-VC gets 68.4% and 46.6% choices for the naturalness and the similarity respectively. For the huge gap between WAStarGAN-VC and StarGAN-VC2 on the naturalness, it might because the W-AdaIN module used in the WAStarGAN-VC model alleviates the information loss, therefore the naturalness has gained an improvement.

However, for the similarity, there are 36.1% of the choices for the option 'can not be decided'. We compute the correlations of the three choices between the naturalness and the similarity. When the naturalness is 'can't be decided', the probability of the similarity being 'can't be decided' is 81.5%. It means that the naturalness might has correlations with the similarity when the naturalness is low.

6. CONCLUSION

In this work, we proposed the WAStarGAN-VC model and tried to achieve StarGAN-based voice conversion under low-resource situations. The subjective and objective evaluation results show that our proposed model has better performance than the baseline models on both naturalness and similarity. Our future work could be one shot voice conversion using StarGAN-based models.

7. REFERENCES

[1] Alexander Kain and Michael W Macon, "Spectral voice conversion for text-to-speech synthesis," in *Proceedings of ICASSP*. IEEE, 1998, vol. 1, pp. 285–288.

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

[3] Srinivas Desai, Alan W Black, B Yegnanarayana, and Kishore 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.

[4] Elias Azarov, Maxim Vashkevich, Denis Likhachov, and Alexander A Petrovsky, "Real-time voice conversion using artificial neural networks with rectified linear units.," in *INTERSPEECH*, 2013, pp. 1032–1036.

[5] Romain Lopez, Jeffrey Regier, Michael I Jordan, and Nir Yosef, "Information constraints on auto-encoding variational bayes," in *NIPS*, 2018, pp. 6114–6125.

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

[7] Ju-chieh Chou, Cheng-chieh Yeh, Hung-yi Lee, and Lin-shan Lee, "Multi-target voice conversion without parallel data by adversarially learning disentangled audio representations," *arXiv preprint arXiv:1804.02812*, 2018.

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *NIPS*, 2014, pp. 2672–2680.

[9] Takuhiro Kaneko and Hirokazu Kameoka, "Parallel-data-free voice conversion using cycle-consistent adversarial networks," *arXiv preprint arXiv:1711.11293*, 2017.

[10] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of ICCV*, 2017, pp. 2223–2232.

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

[12] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proceedings of CVPR*, 2018, pp. 8789–8797.

[13] Takuhiro Kaneko, Hirokazu Kameoka, Kou Tanaka, and Nobukatsu Hojo, "Stargan-vc2: Rethinking conditional methods for stargan-based voice conversion," *arXiv preprint arXiv:1907.12279*, 2019.

[14] Shindong Lee, BongGu Ko, Keonnyeong Lee, In-Chul Yoo, and Dongsuk Yook, "Many-to-many voice conversion using conditional cycle-consistent adversarial networks," in *ICASSP*. IEEE, 2020, pp. 6279–6283.

[15] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur, "A learned representation for artistic style," *arXiv preprint arXiv:1610.07629*, 2016.

[16] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016.

[17] Xun Huang and Serge Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proceedings of ICCV*, 2017, pp. 1501–1510.

[18] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila, "Analyzing and improving the image quality of stylegan," in *Proceedings of CVPR*, 2020, pp. 8110–8119.

[19] Ruobai Wang, Yu Ding, Lincheng Li, and Changjie Fan, "One-shot voice conversion using star-gan," in *Proceedings of ICASSP*. IEEE, 2020, pp. 7729–7733.

[20] Takuhiro Kaneko, Hirokazu Kameoka, Kou Tanaka, and Nobukatsu Hojo, "Cyclegan-vc2: Improved cyclegan-based non-parallel voice conversion," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019.

[21] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha, "Stargan v2: Diverse image synthesis for multiple domains," in *Proceedings of CVPR*, 2020, pp. 8188–8197.

[22] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *ICASSP*. IEEE, 2018, pp. 5329–5333.

[23] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley, "Least squares generative adversarial networks," in *Proceedings of ICASSP*, 2017, pp. 2794–2802.

[24] Junichi Yamagishi, Christophe Veaux, Kirsten MacDonald, et al., "Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit (version 0.92)," 2019.

[25] Masanori Morise, Fumiya Yokomori, and Kenji 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.

[26] Christian Steinmetz, "csteinmetz1/pyloudnorm: 0.1.0 (version v0.1.0)," 2019.

[27] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al., "Pytorch: An imperative style, high-performance deep learning library," in *NIPS*, 2019, pp. 8026–8037.

[28] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[29] Sercan Arik, Jitong Chen, Kainan Peng, Wei Ping, and Yanqi Zhou, "Neural voice cloning with a few samples," in *NIPS*, 2018, pp. 10019–10029.