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

One-shot Voice Conversion with Speaker-agnostic StarGAN

Sefik Emre Eskimez, Dimitrios Dimitriadis, Kenichi Kumatani, Robert Gmyr

Microsoft, One Microsoft Way, Redmond, WA, USA

{sefik.eskimez, dimitrios.dimitriadis, kenichi.kumatani, robert.gmyr}@microsoft.com

Abstract

In this work, we propose a variant of STARGAN for many-to-many voice conversion (VC) conditioned on the d-vectors for short-duration (2-15 seconds) speech. We make several modifications to the STARGAN training and employ new network architectures. We employ a transformer encoder in the discriminator network, and we apply the discriminator loss to the cycle consistency and identity samples in addition to the generated (fake) samples. Instead of classifying the samples as either real or fake, our discriminator tries to predict the categorical speaker class, where a fake class is added for the generated samples. Furthermore, we employ a reverse gradient layer after the generator's encoder and use an auxiliary classifier to remove the speaker's information from the encoded representation. We show that our method yields better results than the baseline method in objective and subjective evaluations in terms of voice conversion quality. Moreover, we provide an ablation study and show each component's influence on speaker similarity.

Index Terms: non-parallel voice conversion, many-to-many voice conversion, one-shot voice conversion, generative adverserial networks, stargan, cyclegan, non-parallel training

1. Introduction

Voice conversion (VC) methods morph the speech characteristics of a source speaker to make it sound similar to the target speaker while retaining the linguistic information. The VC applications include but are not limited to data augmentation for speech processing methods, speaker adaptation for text-to-speech (TTS) methods, de-identification for privacy reasons.

Conventional VC methods typically use paired data for supervised learning between the source and target speakers [1]. However, there are not many paired datasets available, and the existing ones are limited in the number of speakers and samples. It is also costly to collect parallel data due to the precise alignment requirement between source and target speech. As such, it has been proposed to use non-parallel data for the task. However, using non-parallel data for VC is extremely challenging due to the training data mismatch and requires more advanced techniques such as Variational Autoencoders (VAEs) [2, 3], Generative Adversarial Networks (GANs) [4, 5, 6, 7, 8] or careful tuning of the speaker and content encoders [9]. As a result, most of the non-parallel techniques are inferior to the methods trained with parallel data.

Another challenge for VC is the condition used in the neural network for the target speaker. Most of the methods use one-hot speaker vectors as a condition, and the network learns from all audio included in the dataset for the target speaker, which usually varies between 20 minutes and 2 hours. Other methods either use a speaker encoder to extract speaker embeddings or rely on embeddings from systems specifically trained for speaker identification/verification, such as d-vectors.

In this work, we propose a non-parallel VC method using a new variant of STARGAN [7] for short target speech (2-15 seconds). We use d-vectors for conditioning our generator on the target speaker, enabling our method to work for speakers not included in the training set. Our method uses STFT magnitude for input/output and, for fast inference, relies on the Griffin-Lim Algorithm [10] for phase reconstruction. We show that our network yields better performance in terms of VC quality than the baseline methods for experiments with the voice cloning toolkit (VCTK) data.

2. Related Work

VAE-based voice conversion systems have become popular since they do not rely on parallel datasets. Hsu et al. [2] proposed a system that encodes the content of the speech using a VAE to remove any speaker-related information and relies on another latent variable for converting them to the target speaker's style. Saito et al. [3] extended this approach by introducing d-vectors to represent the speaker, allowing this system to perform one-shot VC.

Recently, GAN-based voice conversion methods are getting attention. As such, the most promising GAN-types are CycleGAN [4], STARGAN [7], and STARGAN v2 [11]. While CycleGAN allows unpaired transfer, i.e., not requiring parallel data, between two domains, STARGAN and STARGAN v2 allows unpaired transfer between multiple domains. Chou et al. [12] proposed an autoencoder network trained in two stages. They used an additional generator/discriminator network to generate a residual signal that can improve the converted speech quality.

Kameoka et al. [5] proposed a STARGAN-based VC system for many-to-many voice conversion. Their system uses the mel-cepstral coefficients computed with the WORLD vocoder as input to their network. The predicted features are then converted back to the time domain through their vocoder. The main building-block for the generator and discriminator is a gated CNN [13]. This method needs modification to work with unseen speakers since, during training, they use one-hot speaker labels for style conditioning. Their follow-up work [6] employed instance normalization [14] for infusing the target domain's information. They showed that STARGAN-VC2 outperforms STARGAN-VC. Wang et al. [8] extended STARGAN-VC to be a one-shot voice conversion system by relying on a neural network that extracts the speaker embeddings instead of using one-hot vectors for conditioning the decoder.

As an alternative approach, Kaizhi et al. [9] proposed a system that uses only reconstruction loss. The main idea of their work revolves around hyperparameter tuning for the bottleneck layer's capacity to learn the content representation but not the speaker-related information. Therefore, the decoder relies on the embeddings from the speaker encoder to generate the speaker's style. Their network operates on the mel-spectrograms and relies on Wavenet [15] to generate the wave-

http://dx.doi.org/10.21437/Interspeech.2021-221

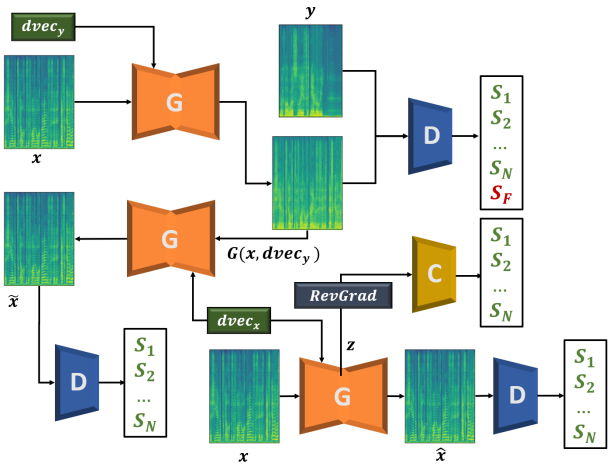

Figure 1: *The system overview for modified STARGAN training is shown. G, D, and C stand for the generator, discriminator, and speaker classifier networks, respectively. x and y are the source and target speech samples. $\hat{x}$ and $\tilde{x}$ are the identity preservation and cycle consistency samples. $dvec_x$ and $dvec_y$ are the d-vectors extracted from x and y, respectively. S_1 to S_N are the speaker IDs, and S_F is the class for the fake samples. $RevGrad$ is the gradient reversal layer, and z is the output of the generator's encoder.*

form. This method is easy to train compared to more complex methods such as GANs; However, due to the usage of Wavenet, the inference is slow. In their follow-up work [16], they conditioned the decoder on the F0 contour to generate more consistent speech with the target speaker's F0 contour. Kaizhi et al. [9] showed that their method yields better results than [7, 12]. Therefore we choose this method as our baseline for performance comparisons. Our contributions and differences from the previous works are as follows:

- We propose a new component for the discriminator's objective function that predicts either the speaker or the fake samples class. We add a class for the fake samples, as done in semi-supervised/data-balancing GANs [17, 18]. This new objective allows us to omit the domain classifier needed by STARGAN, leading to more stable GAN training. Our approach is similar to [19, 20]; however, we use a single class for representing the fake samples.
- We apply the GAN loss to the identity and cycle consistency samples. We show that this approach leads to better speaker similarity when compared to only applying GAN loss to the generated sample.
- A transformer encoder [21] is used for the discriminator.
- We introduce the gradient reversal layer (RevGrad) [22] to ensure suppressing speaker-specific information in the generator's encoder.
- We use short-duration speech between 2 to 15 seconds for extracting the target speaker embedding.

3. Method

Speaker Condition: STARGAN, when applied to VC [5, 6], fails when the target speaker is not included in the training data. A simple modification that replaces the one-hot speaker condition with the speaker embedding condition alleviates this problem [8]. In our approach, we rely on d-vectors for the speaker condition.

Discriminator Objective: In the original STARGAN, the discriminator's objective is to detect if the sample is real or fake while being aware of its domain. Therefore, an additional network (domain classifier) is used to drive the samples to the target domain. In our proposed system, the discriminator uses a semi-supervised/data-balancing GAN [17, 18] loss and does not take any speaker embedding as an input. The acoustic features are fed into the discriminator, and then the discriminator outputs N_{spkr}+1 probabilities, where N_{spkr} is the number of speakers in the training dataset. The last probability represents the fake class. With this modification, the discriminator's new objective is to detect if the sample is generated or if it actually belongs to a speaker. We use categorical cross-entropy to train this network. During the discriminator step, the original speaker labels are used for the real samples, and the fake class labels are used for the generated ones. During the generator step, the original labels are used to drive the generated samples to the target domain.

Discriminator Loss for Identity and Cycle Samples: An identity sample is generated by feeding the generator a d-vector extracted from another sample from the same speaker. Similarly, a cycle sample is created by first converting it to another domain and then converting it back to the original domain. The identity preservation and cycle consistency losses are used to preserve the sample's content, while GAN loss is used to drive the sample's style to the target style. Usually, reconstruction loss is used for these samples since we have the reference point for them. In our system, we propose to use the GAN loss for the identity and cycle samples during the generator's step in order to strengthen the generator's conversion capabilities in addition to the reconstruction loss.

Gradient Reversal Layer: In the generator network, the encoder's job is to extract a content embedding that does not contain any speaker-related information. By reducing the bottleneck layer's dimension (output of the encoder), the information leakage can be reduced. However, it is challenging to tune the bottleneck layer in practice, resulting in speaker information leakage. To further suppress speaker information in the encoder, we propose using the Gradient Reversal Layer (RevGrad)[22] and an additional speaker classification network (C). It is shown that the RevGrad is helpful for speech applications [23]. The training procedure is similar to GAN training: During the generator step, the content embedding is passed through the GradRev layer and fed into the frozen classifier. During the classifier step, the generator is frozen, and the classifier is trained with the speaker embeddings from the generator.

The objective function of our training with respect to G, D, and C is given as follows:

$$\begin{aligned}
\mathcal{L}_G &= \lambda_{GAN}\mathcal{L}^G_{GAN} + \mathcal{L}^G_{rec} + \lambda_{cls} RevGrad(\mathcal{L}^G_{cls}) \\
\mathcal{L}_D &= \mathcal{L}^D_{GAN} \\
\mathcal{L}_C &= \mathcal{L}^C_{cls} \\
\mathcal{L}^G_{GAN} &= \mathcal{L}^G_{GAN}(fake) + \mathcal{L}^G_{GAN}(idt) + \mathcal{L}^G_{GAN}(cyc) \\
\mathcal{L}^G_{rec} &= \lambda_{cyc}\mathcal{L}^G_{cyc} + \lambda_{idt}\mathcal{L}^G_{idt}
\end{aligned} \tag{1}$$

where λ_{GAN}, λ_{cls}, λ_{cyc}, and λ_{idt} are the weight coefficients for the adversarial, classification, cycle consistency and identity preservation losses, respectively.

Generator Network We design our generator network with 1D convolution kernels for fast inference. The generator has

three components: an encoder, a decoder and a d-vector multilayer perceptron (MLP).

Encoder: The encoder's objective is to remove the speaker-related information while keeping the speech content. It contains six convolutional blocks, where each convolutional block contains 1D convolution followed by a LeakyReLU activation then 1D instance normalization [24]. The instance normalization layer helps remove the speaker/domain-specific information [25] and is widely used for image style transfer. The number of filters, filter sizes and stride lengths for these six convolutional blocks are as follows: (256, 7, 1), (256, 5, 2), (128, 3, 1), (128, 3, 2), (64, 3, 1), (64, 3, 2). The number of filters is reduced, and the time dimension is halved every other layer. Since the bottleneck has limited capacity, the encoder cannot capture the complete input information. The encoder's output is fed to the decoder, and during training, it is passed through a RevGrad layer and sent to the speaker classification network to further encourage the removal of the speaker-specific attributes.

D-vector MLP: We employ a four layer MLP for processing the 128-dimensional d-vectors. The hidden dimension for the MLP, which is set to 512, is the same for all hidden layers and output layer. A LeakyReLU activation follows each linear layer. The output of this module is fed to the decoder.

Decoder: The decoder is similar to the encoder: it contains six convolutional blocks and an output block. However, it utilizes adaptive instance normalization layers [26], instead of using instance normalization layers. Each block contains two additional linear layers that take the speaker embedding. The outputs of these two linear layers are the mean and variance for the adaptive instance normalization layer. The number of filters, filter sizes, and upsampling factors for these six convolutional blocks are as follows: (1024, 3, 1), (1024, 3, 2), (512, 3, 1), (512, 3, 2), ($N_{FFT}/2+1$, 5, 1), ($N_{FFT}/2+1$, 5, 2). Instead of transposed convolutions, we employed nearest-neighbor interpolation for upsampling. The output block contains two convolutional layers; the first one is followed by a LeakyReLU activation and has $N_{FFT}/2+1$ filters with a filter size of 7. The second one is followed by a hyperbolic tangent activation and has $N_{FFT}/2+1$ filters with a filter size of 1.

Discriminator Network: Vanilla STARGAN employs a PatchGAN [27] style discriminator, which checks if small patches of images/spectrograms are real or fake. This approach drives the local features to resemble the desired domain. The STARGAN-based VC methods also employ this type of discriminators [5, 6, 8]. Although this approach is practical, we find that changing the local patches without context information (neighboring frames or long-term correlations) can alter the speech's content. Therefore we propose using the transformer encoder [21] as our discriminator. There are six layers in the encoder with eight attention heads and a hidden dimension (H) of 512. We use a bidirectional mask, similar to the BERT model [28], that enables the network to attend all frames of the spectrogram.

The input is first fed to a linear layer projecting the input dimension to the hidden dimension. Then we apply the positional encoding and feed it to the transformer encoder. The last time-step of the output is fed to a linear layer that yields $N_{spkr}+1$ probabilities.

Speaker Embedding Network: We rely on a pre-trained production grade d-vector extractor [29]. It consists of 17 convolutional layers with residual connections. The convolutional layers are used due to their ability to capture local temporal and frequency patterns. We also use a Temporal Average Pooling layer, producing a single embedding from variable-length utterances. The model was trained on the Vox-Celeb corpus [30]. The output d-vectors are normalized to have unit norm. Instead of pooling the d-vectors of the same speaker, we use the d-vectors extracted from a single utterance (2-15 seconds). For short speech segments, the d-vectors can be noisy, allowing the network to generalize better for the unseen d-vectors extracted from a short-duration speech during inference.

Speaker Classification Network: The speaker classification network takes the output of the generator's encoder as an input and outputs the speakers' probabilities. Again, we use a transformer encoder similar to our discriminator. However, we need only two layers since the input features' dimension is 16x smaller than the discriminator input. The other architecture specs are the same. We use categorical cross-entropy to optimize this network, similar to our discriminator.

During the generator step, the speaker classification network is frozen, and only the parameters of the generator's encoder are updated through the gradient reversal layer. During the speaker classification step, the generator is frozen, and we update the parameters of this network only.

4. Experiments

Dataset: In our experiments, we use the voice cloning toolkit (VCTK) dataset. It contains 109 speakers, and each speaker reads around 400 sentences amounting to around 44 hours of speech data, approximately 20-25 minutes per speaker. The sample duration varies between a minimum of 2 seconds and a maximum of 15 seconds. In our experiments, we used the data of 10 speakers for validation, 10 speakers for testing, and 89 speakers for training. Since the testing data does not contain any speakers from the training set, our results reflect our network's performance under unseen speaker conditions. For each sample in the test set, we selected a random speaker that is different from the current speaker among the test set and selected one random utterance from that speaker. Then, we extracted the d-vector of that utterance and used it for converting the voice. All results reported in this section are obtained as such.

Pre-Processing: Our system converts the voices using the short-term Fourier transform (STFT) log-power spectrogram. We extracted the spectral features using 512 FFT bins (N_{FFT}), a 32 ms window-size, and a 16 ms hop-size. We used min-max normalization on the spectrograms, compressing the values to be within the range [-1, 1] to reduce the training complexity. We extracted the minimum and maximum values from the training set and used these statistics both for training and evaluation.

Implementation Details: We implemented our proposed method in the PyTorch framework. For all networks, we used the Adam optimizer with a 1e-4 learning rate, $\beta_1 = 0.5$, and $\beta_2 = 0.99$. The parameter N_{spkr} was 89. We got the best results by setting the weighting coefficients as follows: $\lambda_{GAN} = 1e-3$, $\lambda_{cls} = 1$, $\lambda_{cyc} = 1$, and $\lambda_{idt} = 1$. We selected these values according to the performance of the development set. The networks were trained for 200K iterations. We used spectrograms with 128 time-steps during training, and we set the batch size to 32. We updated all networks in each iteration, and this is critical for convergence. If we use alternating steps, as in vanilla GAN, the system does not converge.

Table 1: *The table shows the objective evaluation results, namely cosine distance and MOSNet predictions, for the baseline and the proposed methods. A lower value is better for the cosine distance, and a higher value is better for the MOSNet score. "IDT-CYC GAN" stands for applying GAN loss to the identity and cycle-consistency samples.*

Model	Rec Only	RevGrad	GAN	IDT-CYC GAN	Cosine Distance	MOSNet Score
AutoVC	✔	N/A	N/A	N/A	0.77	3.46
STARGAN	✖	N/A	✔	N/A	0.79	3.32
Proposed	✔	✖	✖	✖	0.85	3.24
Proposed	✖	✔	✖	✖	0.81	3.37
Proposed	✖	✖	✔	✖	0.71	3.78
Proposed	✖	✖	✔	✔	0.65	4.12
Proposed	✖	✔	✔	✖	0.69	3.82
Proposed	✖	✔	✔	✔	**0.62**	**4.27**

4.1. Evaluation

This section shows our objective and subjective evaluation results for our proposed method and the baseline method[1]. We compare our method with AutoVC-one-shot [9], which was shown to give better results than the STARGAN-VC [7] and Chou et al.'s method [12]. In our experiments, we used the same train, validation, and test data for all methods to ensure a fair comparison. We used the code provided by the author to obtain the baseline method's results. Besides, we employed vanilla STARGAN for comparison against our method. We used the same generator and discriminator architectures as our method, except in this version, discriminator outputs only a single probability for real/fake classification. For the domain classifier, we used the same architecture as our discriminator without the fake class.

Objective Evaluation: Two metrics for objective evaluation are used, cosine distance between d-vectors and the MOSNet [31] score. The cosine distance was calculated as follows: First, we converted each file in the test set by choosing a different random speaker's file and extracted its d-vector. Next, we converted the file and extracted the d-vector from the converted file. Finally, we calculated the cosine distance between the converted and target d-vectors. We repeated this for all files in the test set and used the same file pairs for all methods. MOSNet is a neural network-based mean opinion score (MOS) predictor, confidently correlated with the human MOS, and explicitly developed for evaluating VC models. We used the author's code and pre-trained model trained on Voice Conversion Challenge (VCC) 2018 data [32].

The results are shown in Table 1. The proposed method yields better speaker similarity compared to the baselines. Our ablation study shows that GAN loss plays a more important role than RevGrad loss, yet both yield improvements. Furthermore, we can observe that we can significantly improve the speaker similarity (3.78 to 4.12 and 3.82 to 4.27 MOSNet score improvements) by applying the GAN loss to the identity and cycle samples. The best results are obtained with a combination of all these components. In conclusion, the results suggest that RevGrad can further suppress the speaker information in the generator, and a sequence-based discriminator can drive the generator network to rely more on the d-vector and further push the results to the target speaker.

Subjective Evaluation: We conducted subjective evaluation tests using the mean opinion score (MOS) and ABX tests, following [7], for the baseline and our best performing model. MOS test is designed to estimate the methods' speech quality,

[1]For converted samples: `https://github.com/eeskimez/speaker-agnostic-StarGAN-samples`

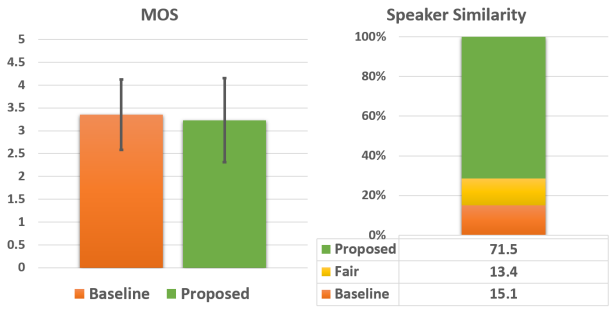

Figure 2: *The subjective evaluation results are shown for the MOS (left) and ABX (right) tests. The baseline system is AutoVC-one-shot.*

where ABX test measures the methods' conversion quality or similarity to the target speaker. We conducted our listening tests on 15 listeners, where each listener was presented with 40 samples for the MOS test and 20 samples for the ABX test. For the MOS test, the samples were randomly presented to the listeners, and they assigned a score between 1 (bad) to 5 (excellent). In the ABX test, "A" and "B" refer to the converted speech samples using the baseline and our proposed methods in these tests, and "X" refers to the real speech sample of the target speaker. The listeners were presented with X, then asked to select either "A" or "B" or "Fair". We randomized the order of A and B to eliminate the order bias.

The results are shown in Figure 2. From these results, our proposed method and baseline method are similar in terms of sound quality, where the baseline is slightly better. This result is not surprising since the baseline relies on Wavenet for waveform construction, where our method relies on Griffin-Lim. Therefore, our samples contain some phase distortion. On the other hand, our method's preference rate is 71.5%, which means our method provides much better speaker similarity, as shown in the right-hand side of Figure 2.

5. Conclusions

In this paper, we presented a method for many-to-many non-parallel voice conversion using short-duration speech. We showed that our method provides better conversion results than the baseline method according to objective and subjective evaluations. Furthermore, our ablation study shows the effect of each proposed component on speaker similarity. One of the limitations of our method is relying on the Griffin-Lim method for reconstructing the signal. This approach can be further improved by methods that include the phase in the generation network.

6. References

[1] B. Sisman, J. Yamagishi, S. King, and H. Li, "An overview of voice conversion and its challenges: From statistical modeling to deep learning," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2020.

[2] 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. IEEE APSIPA*, 2016, pp. 1–6.

[3] Y. Saito, Y. Ijima, K. Nishida, and S. Takamichi, "Non-parallel voice conversion using variational autoencoders conditioned by phonetic posteriorgrams and d-vectors," in *Proc. IEEE ICASSP*, 2018, pp. 5274–5278.

[4] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE ICCV*, 2017, pp. 2223–2232.

[5] 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. IEEE SLT Workshop*, 2018, pp. 266–273.

[6] 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.

[7] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proc. IEEE CVPR*, 2018, pp. 8789–8797.

[8] R. Wang, Y. Ding, L. Li, and C. Fan, "One-shot voice conversion using star-gan," in *Proc. IEEE ICASSP*, 2020, pp. 7729–7733.

[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] D. Griffin and J. Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

[11] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha, "Stargan v2: Diverse image synthesis for multiple domains," in *Proc. IEEE CVPR*, 2020, pp. 8188–8197.

[12] 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," *arXiv preprint arXiv:1804.02812*, 2018.

[13] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *PMLR ICML*, 2017, pp. 933–941.

[14] V. Dumoulin, J. Shlens, and M. Kudlur, "A learned representation for artistic style," *arXiv preprint arXiv:1610.07629*, 2016.

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

[16] K. Qian, Z. Jin, M. Hasegawa-Johnson, and G. J. Mysore, "F0-consistent many-to-many non-parallel voice conversion via conditional autoencoder," in *Proc. IEEE ICASSP*, 2020, pp. 6284–6288.

[17] A. Odena, "Semi-supervised learning with generative adversarial networks," *arXiv preprint arXiv:1606.01583*, 2016.

[18] G. Mariani, F. Scheidegger, R. Istrate, C. Bekas, and C. Malossi, "Bagan: Data augmentation with balancing gan," *arXiv preprint arXiv:1803.09655*, 2018.

[19] S. Lee, B. Ko, K. Lee, I.-C. Yoo, and D. Yook, "Many-to-many voice conversion using conditional cycle-consistent adversarial networks," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6279–6283.

[20] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "Nonparallel voice conversion with augmented classifier star generative adversarial networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2982–2995, 2020.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 5998–6008.

[22] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *PMLR ICML*, 2015, pp. 1180–1189.

[23] Y. Bian, C. Chen, Y. Kang, and Z. Pan, "Multi-reference tacotron by intercross training for style disentangling, transfer and control in speech synthesis," *arXiv preprint arXiv:1904.02373*, 2019.

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

[25] 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.

[26] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proc. IEEE CVPR*, 2017, pp. 1501–1510.

[27] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE CVPR*, 2017, pp. 1125–1134.

[28] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*, 2019, pp. 4171–4186.

[29] T. Zhou, Y. Zhao, J. Li, Y. Gong, and J. Wu, "Cnn with phonetic attention for text-independent speaker verification," in *Proc. IEEE ASRU*, 2019, pp. 718–725.

[30] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: A large-scale speaker identification dataset," *Proc. Interspeech 2017*, pp. 2616–2620, 2017.

[31] C.-C. Lo, S.-W. Fu, W.-C. Huang, X. Wang, J. Yamagishi, Y. Tsao, and H.-M. Wang, "Mosnet: Deep learning based objective assessment for voice conversion," in *Proc. Interspeech 2019*, 2019.

[32] J. Lorenzo-Trueba, J. Yamagishi, T. Toda, D. Saito, F. Villavicencio, T. Kinnunen, and Z. Ling, "The voice conversion challenge 2018: Promoting development of parallel and nonparallel methods," *arXiv preprint arXiv:1804.04262*, 2018.