NVC-NET: END-TO-END ADVERSARIAL VOICE CONVERSION

Bac Nguyen, Fabien Cardinaux

Sony Europe B.V. R&D Center, Stuttgart Laboratory 1

ABSTRACT

Voice conversion (VC) has gained increasing popularity in many speech synthesis applications. The idea is to change the voice identity from one speaker into another while keeping the linguistic content unchanged. Many VC approaches rely on the use of a vocoder to reconstruct the speech from acoustic features, and as a consequence, the speech quality heavily depends on such a vocoder. In this paper, we propose NVC-Net, an end-to-end adversarial network, which performs VC directly on the raw audio waveform. By disentangling the speaker identity from the speech content, NVC-Net is able to perform non-parallel traditional many-to-many VC as well as zero-shot VC from a short utterance of an unseen target speaker. Importantly, NVC-Net is non-autoregressive and fully convolutional, achieving fast inference. Objective and subjective evaluations on VC tasks show that NVC-Net obtains competitive results with significantly fewer parameters.

Index Terms— Voice conversion, adversarial training, end-to-end training, disentangled representation

1. INTRODUCTION

Voice conversion (VC) consists in changing the speech of a source speaker in such a way that it sounds like that of a target speaker while keeping the linguistic information unchanged [1, 2]. Applications of VC include speaker conversions, dubbing in movies, speaking aid systems [3], and pronunciation conversion [4]. Early VC methods require parallel training data, which contain utterances of the same linguistic content spoken by different speakers. However, collecting large parallel corpora and performing time alignment between source and target utterances are often infeasible in practice [5]. This has motivated growing research interest in developing VC methods on non-parallel training data by using auto-encoders [6, 7, 8, 9], generative adversarial networks (GANs) [10, 1, 11, 12, 13] or normalizing flow [14].

Many VC systems reduce the high temporal resolution of the raw audio waveform into lower-dimensional representation (acoustic features). In such a case, one needs a vocoder that reconstructs the waveform representation. Although autoregressive vocoders like WaveNet [15] and WaveRNN [16] can reconstruct good quality audio waveforms, they are slow at inference. Non-autoregressive models like WaveGlow [17] yield faster inference but the waveform reconstruction still dominates the computational effort as well as the memory requirements. Lightweight vocoders like MelGAN [18], Parallel WaveGAN [19], and HiFi-GAN [20] can alleviate the memory issue, however, as observed by Wu et al. [21], the vocoders can produce noisy speech, especially when training data are limited or in the case of a train-test mismatch.

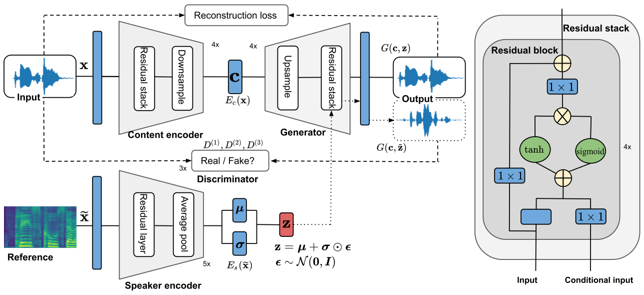


Fig. 1: Overview network architecture of NVC-Net.

While many research efforts have focused on converting audio among speakers seen during training, little attention has been paid to the problem of converting audio from and to unseen speakers (*i.e.*, zero-shot VC). Qian et al. [8] proposed to use a pre-trained speaker encoder on a large data set containing many speakers. With carefully designed bottleneck layers, the content information can be disentangled from the speaker, allowing it to perform zero-shot VC. However, under the context of limited data and computational resources, the speaker encoder cannot generalize well on unseen speakers since the speaker embeddings tend to be scattered [22].

In this paper, we tackle the problem of voice conversion using adversarial training in an end-to-end manner[1]. The main contributions of this paper are three folds as follows. (1) NVC-Net architecture and loss functions aim to disentangle the speaker identity from the speech content. To the best of our knowledge, this is the first GAN-based method that explicitly performs disentanglement for voice conversion directly on the raw audio waveform. (2) NVC-Net can directly generate raw audio without training an additional vocoder. (3) NVC-Net addresses the zero-shot VC problem by constraining the speaker representation. During inference, one can either randomly sample a speaker embedding or extract the speaker embedding from a single reference utterance.

[1] Demo webpage is available at `https://nvcnet.github.io/`

2. NVC-NET

NVC-Net consists of a content encoder E_c, a speaker encoder E_s, a generator G, and three discriminators $D^{(k)}$ for $k = 1, 2, 3$ that are used in different temporal resolutions of inputs. Figure 1 illustrates the overall architecture. We assume that an utterance $\mathbf{x}$ is generated from two latent embeddings, speaker identity $\mathbf{z} \in \mathbb{R}^{d_{\text{spk}}}$ and speech content $\mathbf{c} \in \mathbb{R}^{d_{\text{con}} \times L_{\text{con}}}$, *i.e.*, $\mathbf{x} = G(\mathbf{c}, \mathbf{z})$. The content describes information that is invariant across different speakers, *e.g.*, phonetic and other prosodic information. To convert an utterance $\mathbf{x}$ from speaker y to speaker $\widetilde{y}$ with an utterance $\widetilde{\mathbf{x}}$, we map $\mathbf{x}$ into a content embedding through the content encoder, *i.e.,* $\mathbf{c} = E_c(\mathbf{x})$. In addition, the target speaker embedding $\widetilde{\mathbf{z}}$ is sampled from the output distribution of the speaker encoder $E_s(\widetilde{\mathbf{x}})$. Finally, we generate raw audio from the content embedding $\mathbf{c}$ conditioned on the target speaker embedding $\widetilde{\mathbf{z}}$, *i.e.*, $\widetilde{\mathbf{x}} = G(\mathbf{c}, \widetilde{\mathbf{z}})$.

2.1. Model architecture

Content encoder. The content encoder is a fully-convolutional neural network which can be applied to any input sequence length. The content embedding is at 256x lower temporal resolution than its input. The network consists of 4 downsampling blocks, followed by two convolutional layers with GELU activations [23]. A downsampling block consists of a stack of 4 residual blocks, followed by a strided convolution. A residual block contains dilated convolutions with gated-tanh nonlinearities [15] and residual connections. By increasing the dilation in each block, we aim to capture long-range temporal dependencies of audio signals [15, 18].

Speaker encoder. Given an utterance, the speaker encoder produces a distribution over speaker representation $p(\mathbf{z}|\mathbf{x})$. We assume that $p(\mathbf{z}|\mathbf{x})$ is a conditionally independent Gaussian distribution. The network outputs a mean vector $\boldsymbol{\mu}$ and a diagonal covariance $\boldsymbol{\sigma}^2\mathbf{I}$, where $\mathbf{I}$ is an identity matrix. A speaker embedding is given by sampling from the output distribution, *i.e.*, $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2\mathbf{I})$. The sampling operation can be reparameterized as a differentiable operation using the reparameterization trick [24], *i.e.*, $\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \epsilon$, where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. We simply use mel-spectrograms as inputs to the speaker encoder. The network consists of 5 residual blocks with dilated 1D convolutions. Finally, an average pooling is used to remove temporal dimensions, followed by two dense layers, which output the mean and covariance.

Generator. The network architecture inherits from MelGAN [18], but instead of upsampling pre-computed mel-spectrograms, its input comes from the content encoder, conditioning on the speaker embedding. The network consists of 4 upsampling blocks. Each upsampling is performed by a transposed convolutional layer, followed by a stack of 4 residual blocks with GELU activations [23] and gated-tanh nonlinearities [15]. Finally, the speaker embedding is feed to the residual block as in [15].

Discriminator. Similarly to MelGAN [18], three discriminators with an identical network architecture are applied on different audio resolutions, *i.e.*, downsampled versions of the input with different scales of 1, 2, and 4, respectively. Differently from MelGAN, our discriminator is a multi-task discriminator [25], which contains multiple output branches. Each branch is a binary classification task determining whether an input is a real or a converted utterance.

2.2. Training objectives

Adversarial loss. To make synthesized voices indistinguishable from real voices, the adversarial loss $\mathcal{L}_{\text{adv}}$ is defined as

$$\begin{aligned}\mathcal{L}_{\text{adv}}^{(k)} = \mathbb{E}_{\mathbf{x},y}&[\log(D^{(k)}(\mathbf{x})[y])] \\ &+ \mathbb{E}_{\mathbf{c},\widetilde{\mathbf{z}},\widetilde{y}}[\log(1 - D^{(k)}(G(\mathbf{c}, \widetilde{\mathbf{z}}))[\widetilde{y}])] \,.\end{aligned} \tag{1}$$

The encoders and generator are trained to fool the discriminators, while the discriminators are trained to solve multiple binary classification tasks simultaneously [26]. Each task consists in determining, for one specific speaker, whether an utterance is real or converted. The number of branches is equal to the number of speakers. When updating the discriminator $D^{(k)}$ for an utterance $\mathbf{x}$ of class y, we only penalize it if its y-th branch output $D^{(k)}(\mathbf{x})[y]$ is not correct, while leaving other branch outputs untouched. A reference $\widetilde{\mathbf{x}}$ for a source utterance $\mathbf{x}$ is taken randomly from the same mini-batch.

Reconstruction loss. When generating the audio, we force G to use the speaker embedding by reconstructing the input from the content and speaker embeddings. We use the following feature matching loss [27, 18],

$$\mathcal{L}_{\text{fm}}^{(k)} = \mathbb{E}_{\mathbf{c},\mathbf{z},\mathbf{x}} \sum_{i=1}^{L} \frac{1}{N_i} \| D_i^{(k)}(\mathbf{x}) - D_i^{(k)}(G(\mathbf{c}, \mathbf{z})) \|_1 \,, \tag{2}$$

where $D_i^{(k)}$ denotes the feature map output of N_i units from the discriminator $D^{(k)}$ at the i-th layer, $\|.\|_1$ denotes the ℓ_1-norm, and L denotes the number of layers. To further improve the fidelity of speech, we add the following spectral loss

$$\mathcal{L}_{\text{spe}}^{(w)} = \mathbb{E}_{\mathbf{c},\mathbf{z},\mathbf{x}}[\|\theta(\mathbf{x}, w) - \theta(G(\mathbf{c}, \mathbf{z}), w)\|_2^2] \,, \tag{3}$$

where $\theta(., w)$ computes the log-magnitude of mel-spectrogram with a FFT size of w and $\|.\|_2$ denotes the ℓ_2-norm. The spectral loss is computed at different resolutions $w \in \mathcal{W} = \{2048, 1024, 512\}$.

Content preservation loss. To encourage that the converted utterance preserves the speaker-invariant characteristics of its input audio, we minimize the following loss

$$\mathcal{L}_{\text{con}} = \mathbb{E}_{\mathbf{x},\widetilde{\mathbf{z}}}[\|E_c(\mathbf{x}) - E_c(G(E_c(\mathbf{x}), \widetilde{\mathbf{z}}))\|_2^2]. \tag{4}$$

There are two potential benefits of adding this loss. First, it allows cycle conversion in the sense that if we convert, *e.g.*, an utterance from a speaker A to a speaker B and then convert it back from B to A, we should obtain the original utterance provided that the reconstruction is also minimized. Second, minimizing Eq. (4) also results in disentangling the speaker identity from the speech content. It can be seen that if the content embeddings of utterances from different speakers are the same, the speaker information cannot be embedded in the content embedding. We do not perform any domain classification loss on the content embedding, making the training procedure simpler. Note that the numerical value of ℓ_2-norm in Eq. (4) can be influenced by the scale of the output from E_c. By simply scaling down any $E_c(\mathbf{x})$, the content preservation loss will be reduced. To avoid such situation, we regularize the content embedding to have a unit ℓ_2-norm on the spatial dimension, *i.e.*, $c_{ij} \leftarrow c_{ij}/(\sum_k c_{kj}^2)^{1/2}$.

Kullback-Leibler loss. To perform stochastic sampling from the speaker latent space, we penalize the deviation of the speaker output distribution from a prior zero-mean unit-variance Gaussian, *i.e.*,

$$\mathcal{L}_{\text{kl}} = \mathbb{E}_{\mathbf{x}}[\mathbb{D}_{\text{KL}}(p(\mathbf{z}|\mathbf{x}) \| \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I}))] \,, \tag{5}$$

where $\mathbb{D}_{\text{KL}}$ denotes the Kullback-Leibler (KL) divergence and $p(\mathbf{z}|\mathbf{x})$ denotes the output distribution of $E_s(\mathbf{x})$. Constraining the speaker latent space provides two simple ways to sample a speaker embedding at inference: (i) sample from the prior distribution $\mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I})$ or (ii) sample from $p(\mathbf{z}|\mathbf{x})$ for a reference $\mathbf{x}$. On one hand, this term enforces the speaker embeddings to be smooth and less scattered, making generalization to unseen samples. On the other hand, we implicitly maximize the lower bound approximation of the log-likelihood [24].

Final loss. From Eqs. (1) to (5), the final loss function can be summarized as follows

$$\begin{aligned}\mathcal{L}(E_c, E_s, G) = &\sum_{k=1}^{3} (\mathcal{L}_{\text{adv}}^{(k)} + \lambda_{\text{fm}} \mathcal{L}_{\text{fm}}^{(k)}) + \sum_{w \in \mathcal{W}} \lambda_{\text{spe}} \mathcal{L}_{\text{spe}}^{(w)} \\ &+ \lambda_{\text{con}} \mathcal{L}_{\text{con}} + \lambda_{\text{kl}} \mathcal{L}_{\text{kl}} \,, \\ \mathcal{L}(D) = &- \sum_{k=1}^{3} \mathcal{L}_{\text{adv}}^{(k)} \,,\end{aligned}$$

where $\lambda_{\text{con}} \geq 0$, $\lambda_{\text{fm}} \geq 0$, $\lambda_{\text{spe}} \geq 0$, and $\lambda_{\text{kl}} \geq 0$ control the weights of the objective terms. In our experiments, we set $\lambda_{\text{con}} = 10$, $\lambda_{\text{fm}} = 10$, $\lambda_{\text{spe}} = 10$, and $\lambda_{\text{kl}} = 0.02$.

3. EXPERIMENTS

3.1. Experimental setups

All experiments are conducted on the VCTK data set [28], which contains 44 hours of utterances from 109 speakers of English with various accents, sampled at 22,050 Hz. Utterances from the same speaker are randomly partitioned into training and test sets of ratio 9:1, respectively. NVC-Net is trained with the Adam optimizer using a learning rate of 10^{-4} with $\beta_1 = 0.5$ and $\beta_2 = 0.9$. Inputs to NVC-Net are clips of length 32,768 samples (~1.5 seconds), which are randomly chosen from the utterances.

Following [29], we compute the naturalness and similarity scores of the converted utterances as subjective metrics. The naturalness metric takes into account the amount of distortion and artifacts presented, a score ranging from totally unnatural (1) to totally natural (5). For the similarity metric, the subject is presented with two utterances of the same sentence, one converted and one real, from the target speaker. The subject is explicitly instructed to listen beyond the distortion, misalignment, and mainly to focus on identifying the voice with a score ranging from different speakers (1) to the same speaker (5). The similarity metric aims to measure how well the converted utterance resembles the target speaker identity. There are in total 20 people that participated in our subjective evaluation. We also consider a spoofing assessment as an objective metric. The spoofing measures the percentage of converted utterances being classified as the target speaker. We employ a spectrogram-based classifier trained on the same data split as for training the VC system. A high value indicates that the VC system can successfully convert the utterance to a target speaker, while a low value indicates that the speaker identity is not well captured by the system.

Table 1: Spoofing evaluations of the competing methods

Model	StarGAN-VC2	AutoVC	Blow	NVC-Net†	NVC-Net
Spoofing	19.08	82.46	89.39	**96.43**	93.66

3.2. Voice conversion results on seen speakers

We compare NVC-Net with other state-of-the-art methods in non-parallel VC, including Blow [14], AutoVC [8], and StarGAN-VC2 [12]. All methods use the same training and test sets. No extra data and transfer learning have been used. For AutoVC, we simply use the one-hot encoder for the speaker embeddings. AutoVC uses the WaveNet [15] vocoder pre-trained on the VCTK corpus, which may encode some information of the test utterances in the vocoder. This could be an extra advantage. We also implement another version of our model, called NVC-Net†, which simply uses the one-hot encoder for the speaker embeddings. This allows us to study the importance of the speaker encoder.

The subjective evaluation results are illustrated in Fig. 2. Converted utterances are divided into four groups, including conversions of female to female (F2F), female to male (F2M), male to male (M2M), and male to female (M2F). NVC-Net† and NVC-Net perform well over the four groups as our methods clearly show significant improvement over StarGAN-VC2 and Blow. Both NVC-Net† and NVC-Net outperform the state-of-the-art AutoVC. In terms of naturalness, NVC-Net† yields slightly better than NVC-Net, while both perform equally well in terms of similarity. The spoofing results are shown in Table 1. The classifier used in this experiment achieves a high accuracy of 99.34% on real speech. NVC-Net† gives the highest score with 96.43%, followed by NVC-Net with 93.66%. Both achieve superior performance than other competing methods with a clear margin. Note that the speaker encoder enables zero-shot VC, which makes NVC-Net more useful than NVC-Net† in many applications.

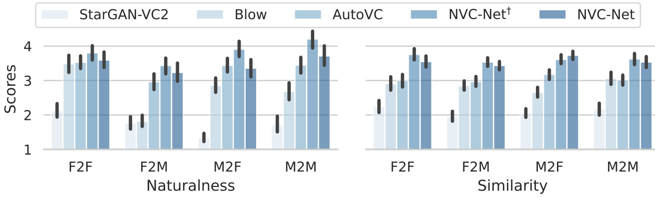


Fig. 2: Subjective evaluation for traditional VC settings with 95% confidence intervals

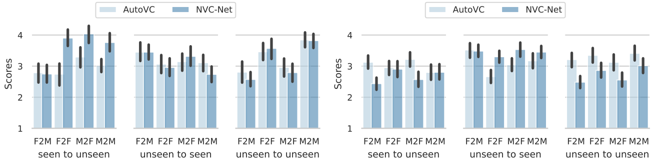


Fig. 3: Subjective evaluation for zero-shot VC settings with 95% confidence intervals (left: naturalness, right: similarity)

3.3. Voice conversion results on unseen speakers

We evaluate the generalization capability of NVC-Net on speakers that are unseen during training. Among the competing methods, only AutoVC supports zero-shot VC. Therefore, we only report the results of NVC-Net against AutoVC. We conduct a similar experiment as in Subsection 3.2. The experiment is split into three different settings, including conversions of seen to unseen, unseen to seen, and unseen to unseen speakers. To get the best results for AutoVC, we use the pre-trained speaker encoder provided by the corresponding authors. The subjective evaluations are shown in Fig. 3. Both NVC-Net and AutoVC achieve competitive results in terms of naturalness. AutoVC gives a slight improvement in terms of similarity. Note that the speaker encoder of AutoVC was trained on a large corpus of 3,549 speakers to make it generalizable to unseen speakers. Clearly, this gives an extra advantage over our method in which no extra data are used.

3.4. Further studies

Disentanglement analysis. The disentanglement of latent representation encoded by the content and speaker encoders is measured as the accuracy of a speaker identification classifier trained and validated on the latent representation. A high value indicates that the latent representation contains the source speaker information. As shown in Table 2, the classification accuracy is high when speaker embeddings are used and it substantially decreases when using content embeddings. This confirms that the speaker encoder is able to disentangle the speaker identity from the content.

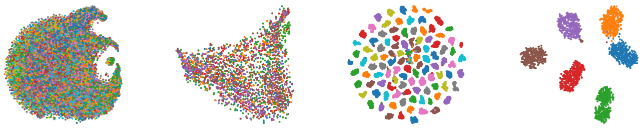
We also analyze NVC-Net without using the content preservation loss in Eq. (4). The speaker classification accuracy increases from 24.15% to 37.33% when the content preservation loss is omitted. This increment implies that the speaker information is also captured in the content code, therefore, we conclude that content preservation loss helps to disentangle the speaker identity from the speech content.

Table 2: Speaker identification accuracy (%) on different embeddings

Model	Content	Speaker
NVC-Net†	19.21	N/A
NVC-Net	24.15	99.22

Table 3: Model size and inference speed comparisons

Model	#params	GPU (kHz)	CPU (kHz)
StarGAN-VC2*	**9.62 M**	60.47	**35.47**
AutoVC*	28.42 M	0.11	0.04
Blow	62.11 M	441.11	2.43
NVC-Net	15.13 M	**3661.65**	7.49



(a) Seen (b) Unseen (c) Seen (d) Unseen

Fig. 4: The Barnes-Hut t-SNE visualization [30] of the content embeddings (see (a) and (b)) and speaker embeddings (see (c) and (d)) of utterances from seen and unseen speakers.

Inference speed and model size. Table 3 shows the inference speed and model size for the competing methods. We did not report the parameters of the vocoders since AutoVC and StarGAN-VC2 can be used together with other choices of vocoders. Compared to the end-to-end Blow method, NVC-Net is significantly smaller, while being faster at inference time. Our model is capable of generating samples at a rate of 3661.65 kHz on an NVIDIA V100 GPU in float precision and 7.49 kHz on a single CPU core of Intel(R) Xeon(R) CPU E5-2695 v3 @ 2.30GHz.

Latent embedding visualization. Figure 4 illustrates the content and speaker embeddings in 2D using the Barnes-Hut t-SNE visualization [30]. Based on speaker embeddings, utterances of the same speaker are clustered together, while those of different speakers are well separated. It is important to note that we do not directly impose any constraints on these speaker embeddings. Due to the KL divergence regularization, the speaker latent space is smooth, allowing to sample a speaker embedding from a prior Gaussian distribution. On the other hand, based on content embeddings, utterances are scattered over the whole space. This indicates that speaker information is not embedded in the content representation.

4. CONCLUSIONS

In this paper, we have introduced NVC-Net, an adversarial neural network which is trained in an end-to-end manner for VC. Rather than constraining the model to work with typical intermediate representation (*e.g.*, spectrograms, linguistic features, etc), NVC-Net is able to exploit a good internal representation through a combination of adversarial feedback and reconstruction loss. In particular, NVC-Net aims to disentangle the speaker identity from the linguistic content. Our empirical studies have confirmed that NVC-Net yields very competitive results in traditional VC settings as well as in zero-shot VC settings. Compared to other VC methods, NVC-Net is more efficient at inference.

5. REFERENCES

[1] T. Kaneko and H. Kameoka, "CycleGAN-VC: Non-parallel voice conversion using cycle-consistent adversarial networks," in *EURASIP*, 2018, pp. 2100–2104.

[2] 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*, vol. 29, pp. 132–157, 2021.

[3] K. Nakamura, T. Toda, H. Saruwatari, and Kiyohiro Shikano, "Speaking-aid systems using gmm-based voice conversion for electrolaryngeal speech," *Speech Communication*, vol. 54, pp. 134–146, 2012.

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

[5] E. Helander, J. Schwarz, J. Nurminen, H. Silen, and M. Gabbouj, "On the impact of alignment on voice conversion performance," in *INTERSPEECH*, 2008, pp. 1453–1456.

[6] C. Hsu, H. Hwang, Y. Wu, Y. Tsao, and H. Wang, "Voice conversion from non-parallel corpora using variational autoencoder," in *APSIPA ASC*, 2016, pp. 1–6.

[7] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "ACVAE-VC: Non-parallel voice conversion with auxiliary classifier variational autoencoder," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, pp. 1432–1443, 2019.

[8] K. Qian, Y. Zhang, S. Chang, X. Yang, and M. Hasegawa-Johnson, "AutoVC: Zero-shot voice style transfer with only autoencoder loss," in *ICML*, 2019, pp. 5210–5219.

[9] Yen-Hao Chen, Da-Yi Wu, Tsung-Han Wu, and Hung-yi Lee, "Again-vc: A one-shot voice conversion using activation guidance and adaptive instance normalization," in *Proceedings at ICASSP*, 2021, pp. 5954–5958.

[10] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "StarGAN-VC: Non-parallel many-to-many voice conversion using star generative adversarial networks," in *Proceedings of the Spoken Language Technology Workshop*, 2018, pp. 266–273.

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

[12] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "StarGAN-VC2: Rethinking conditional methods for stargan-based voice conversion," in *INTERSPEECH*, 2019, pp. 679–683.

[13] Y. Aaron Li, A. Zare, and N. Mesgarani, "Starganv2-vc: A diverse, unsupervised, non-parallel framework for natural-sounding voice conversion," *arXiv preprint arXiv:2107.10394*, 2021.

[14] J. Serrà, S. Pascual, and C. Segura Perales, "Blow: a single-scale hyperconditioned flow for non-parallel raw-audio voice conversion," in *NeurIPS*, 2019, pp. 6793–6803.

[15] A. van den 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," in *INTERSPEECH Workshop*, 2016, pp. 125–125.

[16] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. van den Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," in *ICML*, 2018, pp. 2410–2419.

[17] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in *ICASSP*, 2019, pp. 3617–3621.

[18] 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," in *NeurIPS*, 2019, pp. 14910–14921.

[19] 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 *ICASSP*, 2020, pp. 6199–6203.

[20] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," in *NeurIPS*. 2020, pp. 17022–17033, Curran Associates, Inc.

[21] Y.-C. Wu, K. Kobayashi, T. Hayashi, P. Lumban Tobing, and T. Toda, "Collapsed speech segment detection and suppression for WaveNet vocoder," in *INTERSPEECH*, 2018, pp. 1988–1992.

[22] Y. Saito, S. Takamichi, and H. Saruwatari, "Dnn-based speaker embedding using subjective inter-speaker similarity for multi-speaker modeling in speech synthesis," *arXiv preprint arXiv:1907.08294*, 2019.

[23] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.

[24] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *ICLR*, 2014.

[25] M.-Y. Liu, X. Huang, A. Mallya, T. Karras, T. Aila, J. Lehtinen, and J. Kautz, "Few-shot unsupervised image-to-image translation," in *ICCV*, 2019.

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

[27] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *ICML*, 2016, pp. 1558–1566.

[28] C. Veaux, J. Yamagishi, and K. MacDonald, "VCTK corpus: English multi-speaker corpus for cstr voice cloning toolkit," 2017.

[29] M. Wester, Z. Wu, and J. Yamagishi, "Analysis of the voice conversion challenge 2016 evaluation results," in *INTERSPEECH*, 2016, pp. 1637–1641.

[30] L. van der Maaten, "Accelerating t-SNE using tree-based algorithms," *The Journal of Machine Learning Research*, vol. 15, pp. 3221–3245, 2014.