Flow-VAE VC: End-to-End Flow Framework with Contrastive Loss for Zero-shot Voice Conversion

Le Xu[1,2,*], *Rongxiu Zhong*[2,*], *Ying Liu*[2], *Huibao Yang*[2], *Shilei Zhang*[2]

[1]University of Chinese Academy of Sciences, Beijing, China
[2]China Mobile Research Institute, Beijing, China

xule21@mails.ucas.ac.cn, {zhongrongxiu,liuyingzn,yanghuibao,zhangshilei}@chinamobile.com

Abstract

Voice conversion (VC) seeks to modify one speaker's voice to generate speech as if it came from another speaker. It is challenging especially when source and target speakers are unseen during training (zero-shot VC). Recent work in this area made progress with disentanglement methods that separate utterance content and speaker characteristics from speech audio recordings. However, these models either lack adequate disentanglement ability or rely on the use of a trained vocoder to reconstruct the speech from acoustic features. We propose Flow-VAE VC, which is an end-to-end system processing directly on the raw audio waveform for zero-shot tasks. Flow-VAE VC adopts a conditional Variational Autoencoder (VAE) with normalizing flows and an adversarial training process to improve the expressive power of generative modeling. Specifically, we learn context-invariant representations by applying frame-level contrastive loss to speech different augment samples. The experiments show that the proposed method achieves a decent performance on zero-shot voice conversion and significantly improves converted speech naturalness and speaker similarity. Readers can get the source code and listen to some audio samples on the demo webpage[1].

Index Terms: Voice conversion, VAE, Flow, End-to-end, Zero-shot.

1. Introduction

Voice conversion(VC) aims to change the timbre of one speaker (source speaker) so that it sounds like the timbre of another person (target speaker) while keeping the linguistic content unchanged. It is widely applied in many fields such as entertainment, creative industry, education and healthcare.

According to the training paradigms, there are two major types of methods for VC: parallel and non-parallel [1]. Parallel VC methods trained on parallel data [2] or text transcriptions [3] produce convincing results. Although a VC system with good sound quality can be obtained by training in a supervised way through parallel data, it is often infeasible in practice to collect a large parallel corpus and align the time between the source speech and the target speech. Therefore, non-parallel VC methods solve this problem by learning unlabeled or non-parallel data. Because direct feature mapping method is difficult, a common way to achieve non-parallel VC is disentangling the linguistic and non-linguistic information carried by the source and target utterances, respectively, and training a neural network as a decoder to reconstruct the acoustic feature, with the assumption that the decoder can also generalize well when the linguistic and the non-linguistic information is swapped during the conversion. Various approaches such as generative adversarial network (GAN) based VC [4, 5, 6, 7], variational autoencoder (VAE) based VC [8, 9, 10], and automatic speech recognition (ASR) based VC [11] have been proposed. However, all the methods above can only be used to convert between the limited speakers which are seen in the training dataset.

Recently, zero-shot VC approaches are proposed, which focus on the conversion between the speakers who are unseen in the training dataset. Some approaches such as Auto-vc [12], FragmentVC [13] and VQMIVC [14] employ encoder-decoder frameworks for zero-shot VC, the encoder disentangles the speaking style and content information into the latent embedding, and the decoder generates acoustic features which predefined intermediate (such as mel-spectrogram) by combining both disentangled information. But their training, and inference rely entirely on the output of unsupervised module, this will mix with information other than the content information, and the systems still have to rely on the use of a vocoder to reconstruct the speech from acoustic features, as a consequence, the speech quality heavily depends on a vocoder. To mitigate this problem, a fully end-to-end VC model named NVC-Net [15] which explicitly performs disentanglement for voice conversion directly on the raw audio waveform, but it still lacks of explicit controls over other aspects of speech, e.g., rhythm and prosody. Blow [16] is a normalizing flow network for end-to-end vc on raw audio signals, but it performs many-to-many VC.

Ren [17] observes that VAE is good at capturing the long-range semantics features (e.g., prosody) even with small model size but suffers from blurry and unnatural results; and normalizing flow is good at reconstructing the frequency bin-wise details but performs poorly when the number of model parameters is limited. Inspired by VITS [18], which outperforms the best publicly available TTS systems, we explore a conditional VAE with normalizing flows for VC. In this work, we present an end-to-end system named Flow-VAE VC for zero-shot tasks, Flow-VAE VC adopts a conditional VAE with normalizing flows and an adversarial training process. To obtain a better content representation, we combine the frame-level contrastive loss with information perturbation method which is from NANSY [19]. The main contributions of this paper are three folds as follows:

- Flow-VAE VC is an end-to-end system that does not require parallel data, it performs disentanglement for voice conversion directly on the raw audio waveform and generates raw audio without training an additional vocoder.
- Flow-VAE VC uses the frame-level contrastive loss to speech different augment sample, which can successfully disentangle linguistic content and speaker characteristic.
- Flow-VAE VC addresses the zero-shot VC problem.

[*] Equal contribution.

[1] https://frostmiku.github.io/Flow-VAE-VC/

10.21437/Interspeech.2023-1508

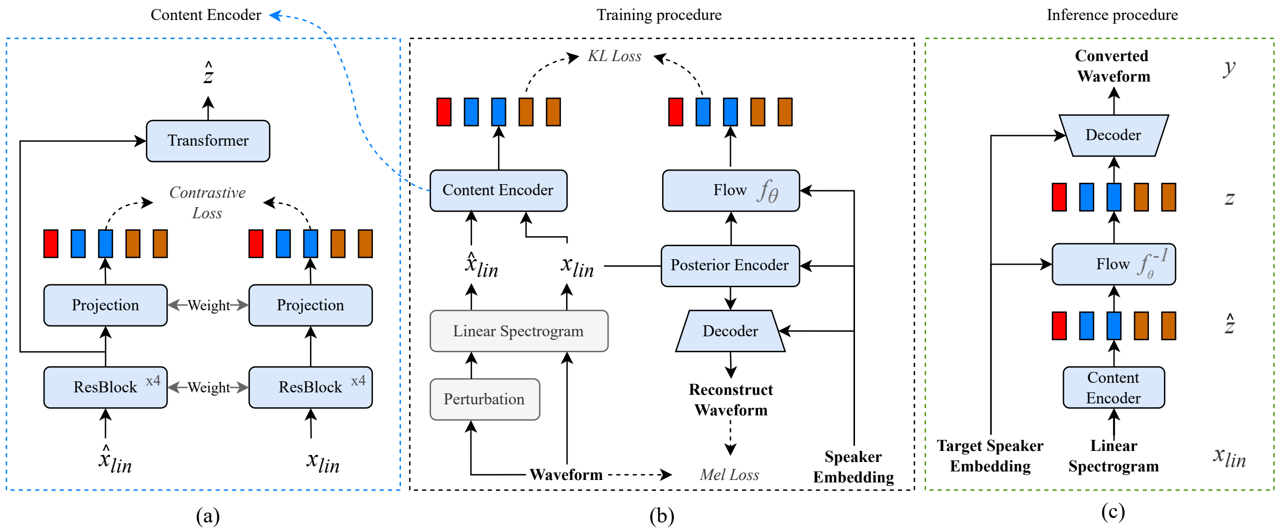

Figure 1: *The architecture of Flow-VAE VC. (a) Latent variables z from the content encoder. (b) Overview training procedure. (c) Voice conversion from the source speaker utterance to target speaker timbre in the inference stage.*

2. Method

2.1. Model Architecture

We describe the main components of Flow-VAE VC, Figure 1 (b) shows an overview of the architecture. The system consists of three components: 1) a content encoder extracts the latent conditional distribution, which is a conditions content representation extracted from the ResNet [20] with contrastive loss. 2) a posterior utterance encoder extracts the latent distribution from the dataset. 3) a decoder that reconstructs the utterance y from z, where z is a sample of the latent distribution.

2.1.1. ELBO

Flow-VAE VC is a conditional VAE model. The optimization objective of the variational autoencoder is to maximize the variational lower bound, which can also be called the evidence lower bound (ELBO) [21].

$$\log p_\theta(x|c) \geq E_{q_\phi(z|x)}\left[\log p_\theta(x|z) - \log \frac{q_\phi(z|x)}{p_\theta(z|c)}\right] \quad (1)$$

where $p_\theta(z|c)$ denotes a content distribution of the latent variables z given condition c, $p_\theta(x|z)$ is the likelihood function of a data point x, and $q_\phi(z|x)$ is an approximate posterior distribution. Maximizing EBLO is equivalent to maximizing $E_{q_\phi(z|x)}\left[\log p_\theta(x|z)\right]$ and minimizing $D_{KL}(q_\phi(z|x)|p_\theta(z|c))$ also can be called construct loss and KL divergence, respectively.

2.1.2. Content encoder

Specifically, as shown in Figure 1 (a), the content encoder is constructed by transformer, which produces the mean and variance from c used for constructing the content distribution z, where c is encoded from the content extractor. To obtain a content representation c, we combine the information perturbation method from NANSY [19] with frame-level contrastive loss. It is mainly based on the fact that this perturbation method only alters the speaker identity of the utterance with minimal changes in the other aspects. Therefore, we designed the frame-level contrastive loss to encourage the model to learn the content representation that is invariant in the source and perturbation to achieve disentanglement of content.

The information perturbation method consists of three components. 1) formant shifting with scaling factor ρ_1. 2) pitch randomization with scaling factor ρ_2. 3) a parametric equalizer is used for random frequency shaping. ρ_1 and ρ_2 is sampled uniformly from $U(1, 1.4)$, and then flipped to their reciprocals with probability of 0.5.

After perturbation, the linear spectrogram of utterance x and its perturbed copy $\hat{x}$ are projected to the embedding space through a pair of content extractors consisting of ResNet with shared weights. To further compute the contrastive loss, we use a Conv1D which out-channels and bias are set to 1 and false respectively to map them to the content code c and $\hat{c}$.

We define the frame-level contrastive loss as:

$$L_{ctr} = -\sum_{i=0}^{T} \log\left[Ctr(i,c,\hat{c}) + Ctr(i,\hat{c},c)\right] \quad (2)$$

where T denotes the number of frames in the content code c and i denote the frame index. $Ctr(\dots)$ is defined as:

$$Ctr(t,c,\hat{c}) = \frac{\exp(d(c_t,\hat{c}_t))}{\sum_{n\in(c\backslash c_t)\cup(\hat{c}\backslash\hat{c}_t)} \exp(d(c_t,n))} \quad (3)$$

where n denotes the frame from a moment other than t as negative samples, $d(c_t, \hat{c}_t)$ denotes calculate the distance between c and $\hat{c}$ at frame t. In this work, we use the cosine angle as a distance metric. To further improve performance, we choose to feed the perturbed copy embedding into the transformer to produce the content distribution. As [18], we apply a normalizing flow f_θ [22] to convert a simple distribution into a more complex distribution. The normalizing flow is a stack of affine coupling layers [23] consisting of a stack of WaveNet [24] residual blocks. we add speaker embedding to the residual blocks in the normalizing flow through global conditioning. the whole of which can be expressed as:

$$p_\theta(z|\hat{x}) = N\left(f_\theta(z);\mu_\theta(\hat{x}),\sigma_\theta(\hat{x})\right)\left|\det\frac{\partial f_\theta(z)}{\partial z}\right| \quad (4)$$

2.1.3. Posterior encoder

The posterior encoder extracts the latent distribution from the linear spectrogram. At the same time, the speaker embedding is also added to the encoding part so that the hidden variable space is more focused on encoding speaker-independent parts. The whole can be expressed as $q_\phi(z|x_{lin}, s_{id})$ where x_{lin} is the linear spectrogram and s_{id} is the speaker embedding extracted from the speaker encoder.

Thus the KL divergence in Eq.1 can be expressed as:

$$\begin{gathered} L_{KL} = \log q_\phi(z \mid w) - \log p_\theta(z \mid \hat{x}) \\ z \sim q_\phi(z \mid w) = N(z; \mu_\phi(w), \sigma_\phi(w)) \\ w = [x_{lin}, s_{id}] \end{gathered} \tag{5}$$

2.1.4. Speaker encoder

In order to achieve any-to-any voice conversion, a robust speaker representation model is essential. We use the pretrained ECAPA-TDNN [25] model to extract the embedding vector s_{id} as the speaker representation, which is trained on Voxceleb1 and Voxceleb2 training data. The pretrained model can be found at Speechbrain [26].

2.1.5. Decoder

The decoder part is consistent with VITS, which is essentially the generator for HiFi-GAN [27]. The decoder upsamples the waveform y from the latent variable z. To achieve any-to-any voice conversion, speaker embedding s_{id} is also added to the latent variable z. Then we transform the original waveform x and y to the mel-spectrogram domain as x_{mel} and y_{mel}, and use the L_1 loss as the reconstructed loss in Eq.6:

$$L_{recon} = |x_{mel} - y_{mel}|_1 \tag{6}$$

2.2. Adversarial training

In order to generate high-quality audio, we introduce adversarial training, and add a discriminator D that distinguishes between the output generated by the decoder G and the ground truth waveform x. And the additional feature matching loss [28] for training the generator:

$$L_{adv}(D) = \mathbb{E}_{(x,z)}\left[(D(x)-1)^2 + (D(G(z)))^2\right] \tag{7}$$

$$L_{adv}(G) = \mathbb{E}_z\left[(D(G(z))-1)^2\right] \tag{8}$$

$$L_{fm}(G) = \mathbb{E}_{(x,z)}\left[\sum_{l=1}^{T} \frac{1}{N_l}||D^l(x) - D^l(G(z))||_1\right] \tag{9}$$

where T denotes the total number of layers in the discriminator and D^l outputs the feature map of the l-th layer of the discriminator with N_l number of features.

2.3. Overall objectives

The total loss for training Flow-VAE VC can be expressed as follows:

$$L = L_{recon} + L_{KL} + L_{ctr} + L_{adv}(G) + L_{fm}(G) \tag{10}$$

2.4. Voice Conversion

In the inference phase, Figure 1 (c) shows the end-to-end voice conversion model, we input the waveform of the target speaker to the speaker encoder for obtaining the target speaker's timbre and then input a linear spectrogram of the source waveform to the content encoder, and the output of the content encoder is sent to the reversed flow module together with the target speaker embedding, and its output is sent to the decoder. Finally, the Flow-VAE VC generates the voice waveform of the target speaker's timbre.

3. Experiments

3.1. Dataset

We conduct our experiment on the VCTK dataset[29], which contains about 46 hours of audio from 109 native English speakers with various accents, and there are about 500 sentences for each speaker. The total length of the audio clips is approximately 44 hours. The audio format is 16-bit PCM with a sample rate of 44 kHz. We reduce the sample rate to 16 kHz and selected 10 speakers as our testing set, where we denote them as our unseen speakers.

3.2. Setup

We use linear spectrograms as input. For the content extractor, we used 4 residual blocks, each consisting of 6 Conv1D layers. We use 80 bands of mel-scale spectrograms for reconstruction loss, which is obtained by applying a mel-filterbank to linear spectrograms.

We use the AdamW optimizer [30] and set $\beta_1 = 0.8$, $\beta_2 = 0.99$ and weight decay $\lambda = 0.01$. We use the Exponential learning rate decay scheduler with a 0.999875 factor in every epoch, where the initial learning rate is set to 0.0002. The seed of the random number generator is set to 1234. We adopt slice training, a method of using only a part of frames for calculating L_{mel} and L_{ctr}, to reduce training time and memory usage during training. We use an NVIDIA V100 GPU with a batch size of 32 and train for 4 days. For more model details of Flow-VAE VC, the code was released on the demo webpage.

To validate our proposed method, we implement comparison and ablation systems. For comparison systems, we selected three SOTA zero-shot VC methods, including FragmentVC [13], VQMIVC [14] and NVC-Net [15]. To make fair comparisons, these models are retrained with the same dataset as Flow-VAE VC. We have set four scenarios for the above systems, including seen2seen, seen2unseen, unseen2seen, and unseen2unseen, where seen2unseen means the source speaker is seen during training and the target speaker isn't. For ablation analysis, we evaluate the effect of the frame-level contrastive loss on the model, Flow-VAE VC w/o Ctr system is composed of speaker and conversion module without applying frame-level contrastive loss to speech different augment samples. Flow-VAE VC is our final proposed system combining frame-level contrastive loss.

3.3. Objective evaluation

We use the mel-cepstral distortion (MCD) [31] to measure how close the converted is to the target speech, and test the CER/WER of the converted speech to evaluate whether the converted voice maintains linguistic content and intonation variations of the source voice. We use the Wenet [32] ASR system to tested the CER and WER of converted audio. We randomly se-

Table 1: *Comparison of evaluated speaker similarity and speaker naturalness MOS on the VCTK dataset. "seen2seen" indicates converting seen speakers to seen speakers, "unseen2seen" means converting unseen speakers to seen speakers, "seen2unseen" means converting seen speakers to unseen speakers, and "unseen2unseen" means converting unseen speakers to unseen speakers.*

Model	FragmentVC	VQMIVC	NVC-NET	Flow-VAE VC w/o Ctr	Flow-VAE VC
Test	Speech naturalness				
seen2seen	2.96	3.36	3.71	3.87	**4.27**
unseen2seen	2.65	3.32	3.49	3.83	**4.25**
seen2unseen	2.48	3.29	3.45	3.80	**4.17**
unseen2unseen	2.10	3.19	3.35	3.76	**4.09**
overall	2.54	3.29	3.50	3.82	**4.12**
Test	Speaker similarity				
seen2seen	3.17	3.57	3.87	3.91	**4.05**
unseen2seen	3.01	3.55	3.75	3.87	**4.01**
seen2unseen	2.85	3.42	3.62	3.75	**3.93**
unseen2unseen	2.61	3.31	3.50	3.62	**3.89**
overall	2.91	3.46	3.69	3.79	**3.97**

lected 4 testing speakers from the testing set as source speakers, and treated the remaining 4 testing speakers as target speakers. The ASR and MCD results of the source speech and the converted speech are shown in Table 2, where we treat the source speech named GT as the topline of all tests. It can be seen that Flow-VAE VC achieves the lowest CER and WER among all methods, which shows the robustness of the proposed method to preserve the source linguistic content. Meanwhile, we observe that the ASR performance of Flow-VAE VC without the contrastive loss (w/o Ctr) is poor, as content information and speaker timbre are not well decoupled. In addition, by providing MCD testing results, it can be seen that Flow-VAE VC methods including without the contrastive loss system generate audio is closer than others.

3.4. Subjective evaluation

For evaluation, we conducted Mean Opinion Score (MOS) tests to evaluate the synthesized results in terms of converted speech naturalness and speaker similarity. The listener needs to give a score for each sample in a test case according to the criterion: 1 = Bad; 2 = Poor; 3 = Fair; 4 = Good; 5 = Excellent. In each test scenario, We randomly select two source speakers and two target speakers from the testing speakers, each source or target set contains one male and one female speaker, which results in 4 conversion pairs, where 20 converted utterances from each pair are evaluated by each subject. The scores are averaged across all pairs and reported in Table 1.

Table 2: *ASR and MCD results.*

Methods	CER(%)	WER(%)	MCD (dB)
GT	3.95	10.16	5.0091
FragmentVC	20.97	33.33	5.2138
VQMIVC	20.44	34.79	5.0773
NVC-NET	32.74	47.10	4.9635
Flow-VAE VC	**19.55**	**30.62**	4.5595
w/o Ctr	29.58	39.13	**4.5288**

Naturalness: As shown in Table 1, the results of naturalness of the converted speech, we can see that our method performs better than the three baseline models in four scenarios, which benefits from the expressive power of generative modeling of Flow-VAE VC. For ablation analysis, the MOS scores of Flow-VAE VC is higher than the proposed model without the contrastive loss in terms of naturalness, which shows the contrastive loss can disentangle speech representation accurately represents linguistic content and then improve converted speech naturalness.

Speaker Similarity: The results of speaker similarity of the converted speech are shown in Table 1, We observe that our method significantly outperforms the three baseline models consistently, in terms of similarity, particularly when source and target speakers are unseen during training. In the experiments of Flow-VAE VC w/o Ctr vs. Flow-VAE VC in the zero-shot voice conversion, the speaker similarity also decreases when the target speaker is unseen, Flow-VAE VC performs better, from our perspective, the contrastive loss can regularize the model to approach an identity mapping when real samples of the target speaker are provided. As the result, we confirm that applying the frame-level contrastive loss is relatively important.

4. Conclusions

In this paper, we propose Flow-VAE VC for raw audio synthesis and specially for the challenging task of zero-shot voice conversion. Flow-VAE VC is an end-to-end system, which adopts a VAE with normalizing flows and an adversarial training process to improve the expressive power of generative modeling and introduces the contrastive loss to the content encoder to enforce the content representation to retain the phonetic structure of the raw speech. Our evaluations showed good results in terms of similarity to target speakers and speech naturalness. We note that although Flow-VAE VC can convert voice with a high level of naturalness, it still lacks control in prosody and needs pretrained model to extract the speaker embedding as conditional input. In the future, we will focus on addressing prosody issue and optimizing speaker encoder.

5. References

[1] 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," in *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018, pp. 195–202.

[2] T. Muramatsu, Y. Ohtani, T. Toda, H. Saruwatari, and K. Shikano, "Low-delay voice conversion based on maximum likelihood estimation of spectral parameter trajectory," 2008.

[3] J.-X. Zhang, Z.-H. Ling, L.-J. Liu, Y. Jiang, and L.-R. Dai, "Sequence-to-sequence acoustic modeling for voice conversion," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 3, pp. 631–644, 2019.

[4] 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.

[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 *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 266–273.

[6] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "Cyclegan-vc3: Examining and improving cyclegan-vcs for mel-spectrogram conversion," *Proc. Interspeech 2020*, pp. 2017–2021, 2020.

[7] ——, "Stargan-vc2: Rethinking conditional methods for stargan-based voice conversion," *Proc. Interspeech 2019*, pp. 679–683, 2019.

[8] 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 *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*. IEEE, 2016, pp. 1–6.

[9] A. Van Den Oord, O. Vinyals *et al.*, "Neural discrete representation learning," *Advances in neural information processing systems*, vol. 30, 2017.

[10] 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, no. 9, pp. 1432–1443, 2019.

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

[12] 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.

[13] Y. Y. Lin, C.-M. Chien, J.-H. Lin, H.-y. Lee, and L.-s. Lee, "Fragmentvc: Any-to-any voice conversion by end-to-end extracting and fusing fine-grained voice fragments with attention," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5939–5943.

[14] D. Wang, L. Deng, Y. T. Yeung, X. Chen, X. Liu, and H. Meng, "Vqmivc: Vector quantization and mutual information-based unsupervised speech representation disentanglement for one-shot voice conversion," 2021.

[15] B. Nguyen and F. Cardinaux, "Nvc-net: End-to-end adversarial voice conversion," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7012–7016.

[16] J. Serrà, S. Pascual, and C. Segura Perales, "Blow: a single-scale hyperconditioned flow for non-parallel raw-audio voice conversion," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[17] Y. Ren, J. Liu, and Z. Zhao, "Portaspeech: Portable and high-quality generative text-to-speech," *Advances in Neural Information Processing Systems*, vol. 34, pp. 13 963–13 974, 2021.

[18] J. Kim, J. Kong, and J. Son, "Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech," in *International Conference on Machine Learning*. PMLR, 2021, pp. 5530–5540.

[19] H.-S. Choi, J. Lee, W. Kim, J. Lee, H. Heo, and K. Lee, "Neural analysis and synthesis: Reconstructing speech from self-supervised representations," *Advances in Neural Information Processing Systems*, vol. 34, pp. 16 251–16 265, 2021.

[20] 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.

[21] M. D. Hoffman and M. J. Johnson, "Elbo surgery: yet another way to carve up the variational evidence lower bound," in *Workshop in Advances in Approximate Bayesian Inference, NIPS*, vol. 1, no. 2, 2016.

[22] D. Rezende and S. Mohamed, "Variational inference with normalizing flows," in *International conference on machine learning*. PMLR, 2015, pp. 1530–1538.

[23] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real nvp," in *International Conference on Learning Representations*.

[24] 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 *9th ISCA Speech Synthesis Workshop*, pp. 125–125.

[25] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: emphasized channel attention, propagation and aggregation in TDNN based speaker verification," in *Interspeech 2020*, H. Meng, B. Xu, and T. F. Zheng, Eds. ISCA, 2020, pp. 3830–3834.

[26] T. Parcollet, M. Ravanelli, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong *et al.*, "Speechbrain: A general-purpose speech toolkit," 2022.

[27] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17 022–17 033, 2020.

[28] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *International conference on machine learning*. PMLR, 2016, pp. 1558–1566.

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

[30] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *ICLR 2019*, 2017.

[31] R. Kubichek, "Mel-cepstral distance measure for objective speech quality assessment," in *Proceedings of IEEE pacific rim conference on communications computers and signal processing*, vol. 1. IEEE, 1993, pp. 125–128.

[32] B. Zhang, D. Wu, C. Yang, X. Chen, Z. Peng, X. Wang, Z. Yao, X. Wang, F. Yu, and L. Xie, "Wenet: Production first and production ready end-to-end speech recognition toolkit," 2021.