StarGAN-VC+ASR: StarGAN-based Non-Parallel Voice Conversion Regularized by Automatic Speech Recognition

Shoki Sakamoto[1], Akira Taniguchi[1], Tadahiro Taniguchi[1], Hirokazu Kameoka[2]

[1]College of Information Science and Engineering, Ritsumeikan University, Japan
[2]NTT Communication Science Laboratories, NTT Corporation, Japan

{sakamoto.shoki, a.taniguchi, taniguchi}@em.ci.ritsumei.ac.jp,
kameoka.hirokazu@lab.ntt.co.jp

Abstract

Preserving the linguistic content of input speech is essential during voice conversion (VC). The star generative adversarial network-based VC method (StarGAN-VC) is a recently developed method that allows non-parallel many-to-many VC. Although this method is powerful, it can fail to preserve the linguistic content of input speech when the number of available training samples is extremely small. To overcome this problem, we propose the use of automatic speech recognition to assist model training, to improve StarGAN-VC, especially in low-resource scenarios. Experimental results show that using our proposed method, StarGAN-VC can retain more linguistic information than vanilla StarGAN-VC.

Index Terms: voice conversion, speech recognition, linguistic information

1. Introduction

Non-parallel many-to-many voice conversion (VC) is a powerful and useful framework for building VC systems [1–3]. VC is the task of converting the para-linguistic characteristics contained in a given speech signal without changing the linguistic content (transcription) [4]. Conventionally, most VC systems have been developed based on a parallel dataset, consisting of pairs of utterances in the source and target domains reading the same sentences [4–6]. However, preparing a large-scale parallel dataset is very expensive, and this poses a potential limitation to both the flexibility and performance of VC systems. Recently, non-parallel VC methods have been proposed and have attracted much attention [7–9]. StarGAN-VC [2] is a non-parallel many-to-many VC method based on a variant of the generative adversarial network (GAN) [10] called StarGAN [11]. This method is particularly attractive in that it can generate converted speech signals fast enough to allow real-time implementation and can generate reasonably realistic sounding speech even with limited training samples.

In StarGAN-VC, the use of cycle consistency and identity mapping losses along with adversarial and domain classification losses encourages the generator to preserve the linguistic content of input speech. However, this function can still fail when the number of available training utterances is extremely small. In such cases, the linguistic content of converted speech can often be corrupted, resulting in inaudible speech. Although non-parallel data are easier to prepare than parallel data, they can be expensive, as in applications such as emotional voice conversion [12, 13]. Hence, there is a need for a method that can overcome this problem that can arise under limited data resources. A possible solution would be to design StarGAN-VC such that it can make explicit use of linguistic information for model training.

VC systems may benefit from being used in combination with automatic speech recognition (ASR) systems owing to the recent advances in ASR performance. One successful example involves the use of phonetic posteriorgrams (PPGs) [14–16]. Similar to the PPG-based approach, the method proposed in this paper improves the ability of StarGAN-VC to preserve the linguistic content of input speech by using ASR to assist model training. Specifically, we propose using the ASR results applied to each training utterance to construct phoneme-dependent regularization terms for the latent vectors generated from the encoder in StarGAN-VC. These penalty terms are derived based on a Gaussian mixture model (GMM); therefore, the latent vectors become more phonetically distinct, resulting in more intelligible speech. We call the proposed method StarGAN-VC+ASR.

The main contributions of this study are as follows. We propose a non-parallel many-to-many VC method (StarGAN-VC+ASR) that combines StarGAN-VC and ASR. We experimentally demonstrate that StarGAN-VC+ASR can produce more intelligible speech than the original StarGAN-VC.

The remainder of this paper is organized as follows. Section 2 briefly introduces StarGAN-VC, which is the basis of the proposed method. Section 3 describes the proposed method, StarGAN-VC+ASR. Section 4 describes an experiment that demonstrates the performance of StarGAN-VC+ASR. Finally, Section 5 concludes the paper.

2. Preliminaries: StarGAN-VC

StarGAN-VC is based on a model consisting of a generator, G, real/fake discriminator, D, and domain classifier, C. Let $o \in \mathbb{R}^{Q \times T}$ be an acoustic feature sequence, where Q is the feature dimension, and T is the length of the sequence. Let $c \in \{1, \ldots, N\}$ be a target domain code, where N is the number of domains. G is a neural network (NN) that takes an acoustic feature sequence, o, in an arbitrary domain and the target domain code, c, as inputs and generates an acoustic feature sequence, $\hat{o}' = G(o, c)$. The network is trained such that $G(o, c)$ becomes indistinguishable from the real speech feature sequence in the domain c. This is made possible by using D and C to train G, whose roles are to discriminate fake samples, $G(o, c)$, from real samples, o', and identify the class to which $G(o, c)$ is likely to belong. Figure 1 shows the overview of StarGAN-VC training. D is designed to produce a sequence of probabilities $D(o', c)$, each of which indicates the likelihood that a different segment of input o' is real, whereas C is designed to produce a sequence of class probabilities $p_C(c|o')$, each of which indicates the likelihood that a different segment of o' belongs to a particular class.

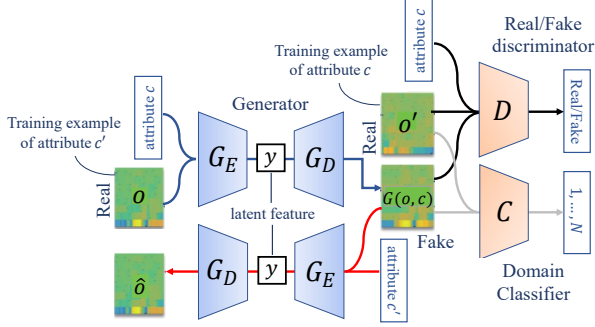For training, StarGAN-VC uses adversarial, cycle consis-

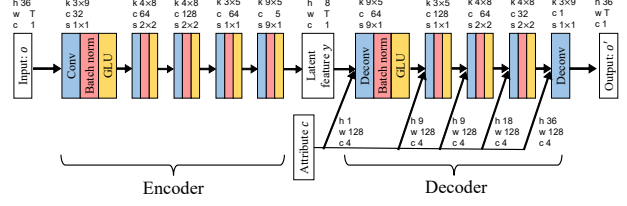Figure 1: *Concept of StarGAN-VC training. G_E and G_D represent the encoder and decoder in G, respectively.*

Figure 2: *The network architecture of generator G. In input and latent features, "h," "w," and "c" represent height, width, and number of channels, respectively. In addition, "k," "c," and "s" denote kernel size, number of channels, and stride size, respectively. "Conv," "Batch norm," "GLU," "Deconv," "Sigmoid," "Softmax," and "Product" denote convolution, batch normalization, gated linear unit, transposed convolution, sigmoid, softmax, and product pooling layers, respectively.*

tency, identity mapping, and domain classification losses.

Adversarial Loss: The adversarial losses are defined for discriminator D and generator G as

$$\mathcal{L}_{\text{adv}}^D(D) = -\mathbb{E}_{c \sim p(c), o' \sim p(o'|c)} \left[\log D(o', c) \right]$$
$$- \mathbb{E}_{o \sim p(o), c \sim p(c)} \left[\log(1 - D(G(o, c), c)) \right], \quad (1)$$

$$\mathcal{L}_{\text{adv}}^G(G) = -\mathbb{E}_{o \sim p(o), c \sim p(c)} \left[\log D(G(o, c), c) \right], \quad (2)$$

where $o' \sim p(o'|c)$ denotes a training sample of acoustic feature sequences of real speech belonging to domain c, and $o \sim p(o)$ denotes an arbitrary domain. Equation 1 takes a small value when D correctly identifies $G(o, c)$ as fake and o' as real. By contrast, Equation 2 takes a small value when D misclassifies $G(o, c)$ as real. Thus, the subgoals of D and G are to minimize Equations 1 and 2, respectively.

Domain Classification Loss: The domain classification losses for classifier C and generator G are defined as

$$\mathcal{L}_{\text{cls}}^C(C) = -\mathbb{E}_{c \sim p(c), o' \sim p(o'|c)} \left[\log p_C(c|o') \right], \quad (3)$$

$$\mathcal{L}_{\text{cls}}^G(G) = -\mathbb{E}_{o \sim p(o), c \sim p(c)} \left[\log p_C(c|G(o, c)) \right]. \quad (4)$$

Equations 3 and 4 have small values when $o' \sim p(o'|c)$ and $G(o, c)$ are correctly classified as belonging to domain c by C. Thus, the subgoals of C and G are to minimize Equations 3 and 4, respectively.

Cycle-Consistency Loss: The adversarial and classification losses encourage $G(o, c)$ to become realistic and classifiable, respectively. However, using these losses alone does not guarantee that G preserves the linguistic content of the input speech. To promote content-preserving conversion, the cycle consistency loss

$$\mathcal{L}_{\text{cyc}}(G)$$
$$= \mathbb{E}_{c' \sim p(c), o \sim p(o|c'), c \sim p(c)} \left[||G(G(o, c), c') - o||_\rho \right], \quad (5)$$

is used, where $o \sim p(o|c')$ denotes a training sample of acoustic feature sequences of real speech belonging to the source domain c', and ρ is a positive constant.

Identity-Mapping Loss: To ensure that G keeps its input, o, unchanged if o already belongs to the target domain, the identity mapping loss is defined as

$$\mathcal{L}_{\text{id}}(G) = \mathbb{E}_{c' \sim p(c), o \sim p(o|c')} \left[||G(o, c') - o||_\rho \right]. \quad (6)$$

The full loss function is given as

$$\mathscr{I}_G(G) = \mathcal{L}_{\text{adv}}^G(G) + \lambda_{\text{cls}} \mathcal{L}_{\text{cls}}^G(G)$$
$$+ \lambda_{\text{cyc}} \mathcal{L}_{\text{cyc}}(G) + \lambda_{\text{id}} \mathcal{L}_{\text{id}}(G), \quad (7)$$

$$\mathscr{I}_D(D) = \mathcal{L}_{\text{adv}}^D(D), \quad (8)$$

$$\mathscr{I}_C(C) = \mathcal{L}_{\text{cls}}^C(C), \quad (9)$$

$\lambda_{cls} \geq 0$, $\lambda_{cyc} \geq 0$, and $\lambda_{id} \geq 0$ are regularization parameters that weigh the importance of the domain classification, cycle-consistency, and identity-mapping losses.

Regarding network architectures, we assume an encoder-decoder type architecture for G, as shown in Figure 2. The encoder is responsible for extracting a speaker-independent latent feature sequence y, which is expected to correspond to the linguistic content of the input utterance. The decoder, on the other hand, is responsible for reconstructing an acoustic feature sequence using both target domain codes c and y.

3. StarGAN-VC+ASR

Here, we present the idea of StarGAN-VC+ASR. Figure 3 shows the overview of the StarGAN-VC+ASR training.

3.1. GMM-based phoneme model in latent space

G has an encoder-decoder structure similar to variational autoencoders (VAEs) [17]. The encoder part in G is responsible for extracting a speaker-independent latent feature sequence, y, corresponding to the linguistic content of the input. Now, to associate each element of y with the phoneme in the corresponding frame of the input speech, we consider constructing a prior distribution over y based on the result of applying ASR to that speech in advance. Let $P = \{1, 2, \ldots, K\}$ denote a set of phoneme indices, where K denotes the number of phonemes. In the ASR step, the input is the acoustic feature sequence of dth utterance, denoted as $x^d = x_{1:T_d}^d = (x_1^d, x_2^d, \ldots, x_{T_d}^d)$, where T_d and t denote the frame length and index, respectively. The ASR output is assumed to be a sequence of phoneme indices $\hat{z}^d = \hat{z}_{1:T_d}^d = \text{ASR}(x_{1:T_d}^d)$ of the same length. The encoder part in G takes the acoustic feature sequence o^d of the same utterance as the input and produces a latent feature sequence $y^d = y_{1:T_d}^d = (y_1^d, y_2^d, \ldots, y_{T_d}^d)$. To ensure that the latent features corresponding to the same phoneme are close to each other, we assign a single Gaussian distribution to each phoneme index and model
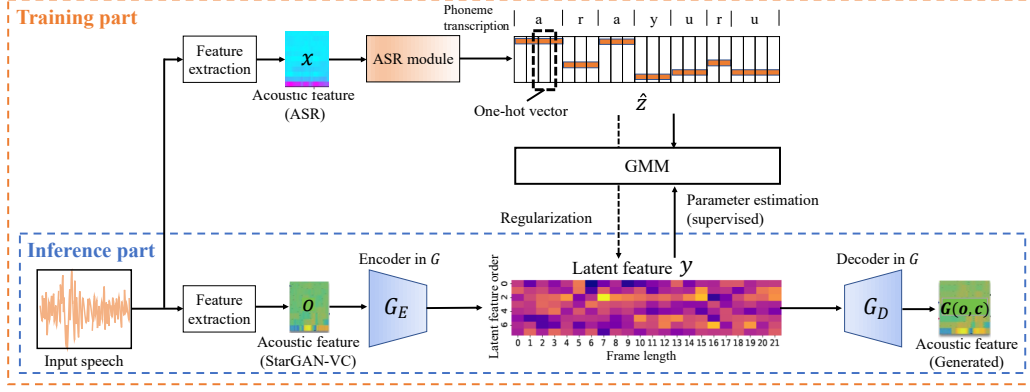
Figure 3: *Overview of StarGAN-VC+ASR training*

a prior distribution of y^d according to $\hat{z}^d$. Given y^d and $\hat{z}^d$, the maximum likelihood estimates of the mean and covariance of each Gaussian can be obtained as

$$\mu_p = \frac{1}{\sum_{d=1}^{D} \#(T_p(\hat{z}^d))} \sum_{d=1}^{D} \sum_{t \in T_p(\hat{z}^d)} y_t^d, \tag{10}$$

$$\Sigma_p = \frac{1}{\sum_{d=1}^{D} \#(T_d(\hat{z}^d))} \sum_{d=1}^{D} \sum_{t \in T_p(\hat{z}^d)} \left(y_t^d - \mu_p\right) \left(y_t^d - \mu_p\right)^T, \tag{11}$$

where p and D denote the phoneme index in P and the number of training utterances, respectively. $T_p(\hat{z}^d)$ denotes the set consisting of t that satisfies $\hat{z}_t^d = p$, and $\#(\cdot)$ denotes the number of elements in a set. In the following, we use μ_P and Σ_P to denote the sets $\mu_P = \{\mu_1, \mu_2, \ldots, \mu_{\#(P)}\}$ and $\Sigma_P = \{\Sigma_1, \Sigma_2, \ldots, \Sigma_{\#(P)}\}$, respectively.

3.2. Phoneme-dependent regularization loss

The training process of StarGAN-VC+ASR consists of two stages. The first stage corresponds to the original StarGAN-VC training described in Section 2. In the second stage, G is updated by using μ_P and Σ_P obtained by Equations 10 and 11 for regularization. Specifically, we assume

$$p(y_t^d) = \mathcal{N}\left(y_t^d \mid \mu_{\hat{z}_t^d}, \Sigma_{\hat{z}_t^d}\right), \tag{12}$$

as the prior distribution of y^d, and define

$$\mathcal{L}_{\text{ASR}}(G) = -\sum_{d=1}^{D} \sum_{t} \log p(y_t^d) \tag{13}$$

as the regularization term for the second stage of the training. Although very simple, this regularization is expected to bring each element of y^d corresponding to the same phoneme closer to each other.

To summarize, the loss function of the generator in StraGAN-VC+ASR $\mathcal{I}_G(G)$ is denoted as

$$\begin{aligned} \mathcal{I}_G(G) = &\mathcal{L}_{\text{adv}}^G(G) + \lambda_{\text{cls}} \mathcal{L}_{\text{cls}}^G(G) \\ &+ \lambda_{\text{cyc}} \mathcal{L}_{\text{cyc}}(G) + \lambda_{\text{id}} \mathcal{L}_{\text{id}}(G) \\ &+ \beta \mathcal{L}_{\text{ASR}}(G), \end{aligned} \tag{14}$$

where $\beta \geq 0$ is the hyperparameter that weighs the importance of the regularization term. In addition to the above, the loss functions for D and C are the same as those in Equations 8 and 9, respectively.

4. Experiments

4.1. Experimental setup

Dataset: We evaluated our method on a multi-speaker VC task using an ATR digital sound database [18], which consists of recordings of five female and five male Japanese speakers. The spoken sentences were recorded as waveforms and were sampled at 20 kHz. We used a subset of speakers for training and evaluation. We selected two female speakers, "FKN" and "FTK," and two male speakers, "MMY" and "MTK", resulting in twelve different combinations of source and target speakers. We selected 24 and 20 sentences for training and evaluation, respectively. Therefore, there were $12 \times 20 = 240$ test signals in total.

Conversion process: 36 Mel-cepstral coefficients (MCEPs), logarithmic fundamental frequency (log F_0), and aperiodicities were extracted every 5 ms using the WORLD analyzer [19] (D4C edition [20]). In these experiments, we applied the baseline and proposed methods only to MCEP conversion. The F_0 contours were converted by logarithmic Gaussian normalization [21]. The aperiodicities were used directly without modification. We trained the networks using the Adam optimizer [22] with a batch size of 8. The number of iterations was set to 2×10^3, the learning rates for G and D were set to 0.001, and the momentum term was set to 0.5. We set $\lambda_{cls} = 1.0$, $\lambda_{cyc} = 1.0$, $\lambda_{id} = 1.0$ following the original paper [2]. We set $\beta = 0.01$ empirically.

Implementation details: We used a dictation-kit (version 4.5)[1] as the trained ASR system. This package contains executables of Julius (deep neural network (DNN) version) [23], Japanese acoustic models (AM), and Japanese language models (LM). Julius is a high-performance, small-footprint, large-vocabulary continuous speech recognition (LVCSR) decoder software used by speech-related researchers and developers. The AM is a speaker-independent triphone DNN-hidden Markov model (H-MM) trained using the JNAS and CSJ corpora. It also has regression tree classes that are required for speaker adaptation by HTK[2]. The LMs are 60k-word N-gram language models trained using the BCCWJ corpus. Moreover, we set the dictionary data, which consisted of only utterances of training data, to improve the accuracy of speech recognition.

---

[1] https://osdn.net/dl/julius/dictation-kit-4.5.zip
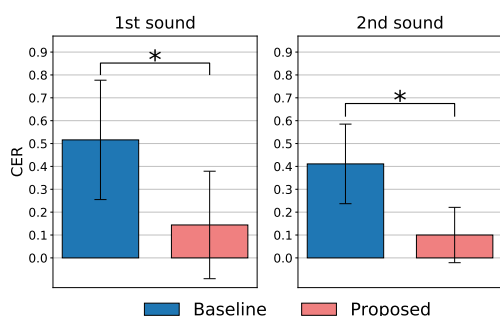[2] Hidden Markov Model Toolkit (HTK), http://htk.eng.cam.ac.uk/

Figure 4: *Character error rates (CERs) for retention of linguistic information. CERs of the 1st and 2nd sound represent the first and second sounds heard by the listeners, respectively, and the content uttered are the same. The asterisks, *, indicate that the coefficients are statistically different from zero at the 1 percent level.*
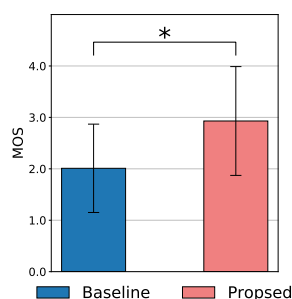

Figure 5: *Result of the MOS test for naturalness. The asterisks, *, indicate that the coefficients are statistically different from zero at the 1 percent level.*

4.2. Subjective evaluation

We conducted listening tests to analyze the performance of the proposed method compared to StarGAN-VC [2], which is the baseline in our proposed method. To measure the consistency of the linguistic contents of input and converted speech, we used the character error rate (CER) test as an evaluation metric. CER evaluates the error rate between the correct sentence and the sentence written by a listener after listening to the converted speech. During the CER test, the listeners hear the same utterance of a converted speech twice because they may not be able to hear the converted speech at once. The first trial is denoted as "1st sound," and the second trial is denoted as "2nd sound" in Figure 4. To measure naturalness, we conducted a mean opinion score (MOS) test. During the MOS test, listeners were asked to rate the naturalness of the converted speech on a 5-point scale. To measure the clarity of linguistic information, we conducted an AB test. "A" and "B" were the speech converted by the baseline and proposed methods, respectively. For each sentence pair, the listeners were asked to select their preferred one, "A," "B," or "Fair." We conducted CER and MOS tests as in Experiment 1, and the AB test as in Experiment 2. Ten sentences were randomly selected for Experiments 1 and 2 from 240 synthesized speech signals. Sentences of different utterances were selected between Experiment 1 and Experiment 2. Sixteen well-educated Japanese speakers participated in the tests.
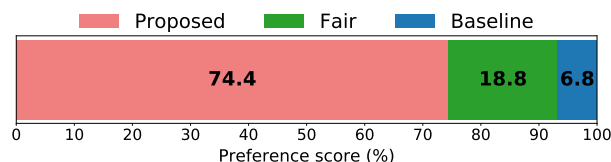

Figure 6: *Results of the AB test for the retention of linguistic information*

Results of Experiment 1: Figure 4 and Figure 5 show the average CER score for the retention of linguistic information and MOS for naturalness, respectively. As shown by the results of the CER test, the proposed method significantly outperformed the baseline method in terms of retaining the linguistic information of both 1st and 2nd sounds. In addition, the MOS test shows that the proposed method outperformed the baseline method in terms of naturalness.

Result of Experiment 2: Figure 6 shows the score of the AB test in terms of the clarity of linguistic information. Based on the score, the proposed method achieved the highest score in terms of the clarity of linguistic information.

5. Conclusion

The original StarGAN-VC can fail to preserve the linguistic content of input speech when the number of training utterances is very small. To address this issue, in this paper, we proposed StarGAN-VC+ASR, a method that exploits the result of ASR applied to each training utterance during the training of StarGAN-VC's generator. Experimental results showed that StarGAN-VC+ASR had a better ability to preserve linguistic contents than the original StarGAN-VC.

In the present experiment, we used a small dataset for training and showed that our method outperformed StarGAN-VC. However, as the size of the dataset increases, the performance of StarGAN-VC is also expected to improve. In the future, we plan to investigate how the performance of StarGAN-VC and StarGAN-VC+ASR changes as the size of the dataset increases.

Moreover, to improve the quality of sounds converted by the proposed method, employing a mel-spectrogram and neural network vocoder [24–26] instead of the MCEPs and WORLD analyzer, respectively, can be a future challenge.

In this study, we applied StarGAN-VC+ASR to speaker-identity voice conversion. However, the application of non-parallel voice conversion is not limited to speaker-identity voice conversion. For example, emotional voice conversion is an important target for voice conversion. The application of StarGAN-VC+ASR to emotional voice conversion will be our future work. StarGAN-VC+ASR used phoneme index information as a regularization term. This term improved the clarity of the converted voice. However, this may reduce the diversity of para-linguistic expressions in utterances and negatively affect emotional voice conversion. Validating this point will also be part of our future work.

6. Acknowledgements

7. References

[1] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, "Neural discrete representation learning," in *Proc. Conference on Neural Information Processing Systems (NIPS)*, 2017.

[2] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "Stargan-vc: Non-parallel many-to-many voice conversion using star generative adversarial networks," in *IEEE Spoken Language Technology Workshop (SLT)*, 2018, pp. 266–273.

[3] K. Qian, Y. Zhang, S. Chang, X. Yang, and M. Hasegawa-Johnson, "Autovc: Zero-shot voice style transfer with only autoencoder loss," in *Proc. International Conference on Machine Learning*, 2019, pp. 5210–5219.

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

[5] S. Desai, A. W. Black, B. Yegnanarayana, and K. 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.

[6] T. Kaneko, H. Kameoka, K. Hiramatsu, and K. Kashino, "Sequence-to-sequence voice conversion with similarity metric learned using generative adversarial networks." in *Proc. The Annual Conference of the International Speech Communication Association (Interspeech)*, 2017, pp. 1283–1287.

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

[8] Hsu, Chin-Cheng and Hwang, Hsin-Te and Wu, Yi-Chiao and Tsao, Yu and Wang, Hsin-Min, "Voice conversion from unaligned corpora using variational autoencoding wasserstein generative adversarial networks," in *Proc. The Annual Conference of the International Speech Communication Association (Interspeech)*, 2017, pp. 3364–3368.

[9] T. Kaneko and H. Kameoka, "Cyclegan-vc: Non-parallel voice conversion using cycle-consistent adversarial networks," in *Proc. 26th European Signal Processing Conference (EUSIPCO)*, 2018, pp. 2100–2104.

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

[11] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5967–5976.

[12] G. Rizos, A. Baird, M. Elliott, and B. Schuller, "Stargan for emotional speech conversion: Validated by data augmentation of end-to-end emotion recognition," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 3502–3506.

[13] A. Moritani, R. Ozaki, S. Sakamoto, H. Kameoka, and T. Taniguchi, "Stargan-based emotional voice conversion for japanese phrases," *arXiv preprint arXiv:2104.01807*, 2021.

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

[15] T. Kinnunen, L. Juvela, P. Alku, and J. Yamagishi, "Non-parallel voice conversion using i-vector plda: Towards unifying speaker verification and transformation," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 5535–5539.

[16] T. Xiaohai, W. Zhichao, Y. Shan, Z. Xinyong, D. Hongqiang, Z. Yi, Z. Mingyang, Z. Kun, S. Berrak, X. Lei, and L. Haizhou, "The NUS & NWPU system for voice conversion challenge 2020," in *Proc. Joint Workshop for the Blizzard Challenge and Voice Conversion Challenge*, 2020, pp. 170–174.

[17] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv*, vol. abs/1312.6114, 2013.

[18] A. Kurematsu, K. Takeda, Y. Sagisaka, S. Katagiri, H. Kuwabara, and K. Shikano, "Atr japanese speech database as a tool of speech recognition and synthesis," *Speech Communication*, vol. 9, no. 4, pp. 357–363, 1990.

[19] M. Morise, F. Yokomori, and K. 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.

[20] M. Morise, "D4c, a band-aperiodicity estimator for high-quality speech synthesis," *Speech Communication*, vol. 84, pp. 57–65, 2016.

[21] K. Liu, J. Zhang, and Y. Yan, "High quality voice conversion through phoneme-based linear mapping functions with straight for mandarin," in *Proc. IEEE Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, vol. 4, 2007, pp. 410–414.

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 2014.

[23] A. Lee and T. Kawahara, "Recent development of open-source speech recognition engine julius," in *Proc. Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2009, pp. 131–137.

[24] A. Tamamori, T. Hayashi, K. Kobayashi, K. Takeda, and T. Toda, "Speaker-dependent wavenet vocoder." in *Proc. The Annual Conference of the International Speech Communication Association (Interspeech)*, vol. 2017, 2017, pp. 1118–1122.

[25] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 3617–3621.

[26] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brébisson, Y. Bengio, and A. Courville, "Melgan: Generative adversarial networks for conditional waveform synthesis," *Proc. Conference on Neural Information Processing Systems (NIPS)*, pp. 14 881–14 892.