Multi-target Voice Conversion without Parallel Data by Adversarially Learning Disentangled Audio Representations

Ju-chieh Chou[1], Cheng-chieh Yeh[1], Hung-yi Lee[1], Lin-shan Lee[1]

[1]College of Electrical Engineering and Computer Science, National Taiwan University

{r06922020,r06942067,hungyilee}@ntu.edu.tw, lslee@gate.sinica.edu.tw

Abstract

Recently, cycle-consistent adversarial network (Cycle-GAN) has been successfully applied to voice conversion to a different speaker without parallel data, although in those approaches an individual model is needed for each target speaker. In this paper, we propose an adversarial learning framework for voice conversion, with which a single model can be trained to convert the voice to many different speakers, all without parallel data, by separating the speaker characteristics from the linguistic content in speech signals. An autoencoder is first trained to extract speaker-independent latent representations and speaker embedding separately using another auxiliary speaker classifier to regularize the latent representation. The decoder then takes the speaker-independent latent representation and the target speaker embedding as the input to generate the voice of the target speaker with the linguistic content of the source utterance. The quality of decoder output is further improved by patching with the residual signal produced by another pair of generator and discriminator. A target speaker set size of 20 was tested in the preliminary experiments, and very good voice quality was obtained. Conventional voice conversion metrics are reported. We also show that the speaker information has been properly reduced from the latent representations.

Index Terms: voice conversion, disentangled representation, adversarial training.

1. Introduction

Speech signals inherently carry both linguistic and acoustic information. Voice conversion (VC) aims to convert the speech signals from a certain acoustic domain to another while keeping the linguistic content unchanged. Examples of acoustic domains may include speaker identity [1, 2, 3, 4, 5, 6], speaking style, accent, emotion [7] or some other properties orthogonal to the linguistic content. Voice conversion (VC) can be used for various tasks such as speech enhancement [8, 9], language learning for non-native speakers [7], to name a few [10]. This work focuses on the conversion of speaker identity.

In general, among the difficult problems for VC approaches, the need of aligned data [11, 12, 13, 14, 15, 16, 17], and over-smoothing of signals [1, 18, 19] are two examples carefully studied. Due to the difficulties in obtaining aligned corpora, approaches utilizing generative models such as Variational Autoencoders (VAEs) [20] and Generative Adversarial Networks (GANs) [21] were studied because they can be trained with non-parallel data. With VAEs, the encoder learns the speaker-independent linguistic content, which can then be used by the decoder to generate the voice with a specified speaker id [22, 23, 24]. Cycle-consistent adversarial network (Cycle-GAN) was also used to learn the mapping from the source speaker to the target speaker in an unsupervised way [25, 26].

Some prior works successfully used VAEs for VC, but generated the voice frame-by-frame [23]. Some other prior works were able to disentangle the linguistic content from the speaker characteristics when learning the representations, but based on some heuristic assumptions [24]. Cycle-GAN was used for VC without parallel data, but an individual model is needed for each target speaker [25, 26]. In this paper, we propose an autoencoder architecture which is able to deal with several frames at a time, leading to better results because in this way information carried by neighboring frames can be considered. This approach also uses jointly trained speaker classifier to remove the need for heuristic assumptions made previously. This approach is able to train a single model to convert the voice to many different speakers, all without parallel data, by separating the speaker characteristics from the linguistic content. This is similar to some degree to some works in computer vision which learned disentangled representation [27], or shared generator with conditional input [28].

The proposed approach includes two stages of training as in Fig. 1 [1]. In stage 1, we train an autoencoder. The encoder encodes the input spectra into a latent representation for the linguistic content but without speaker characteristics based on the adversarial training concept. This is achieved by training a classifier to classify the speaker based on the latent representation, while the encoder is trained adversarially to fool the classifier. On the other hand, the decoder merges the speaker identity with the latent representation to reconstruct the original spectra.

In stage 2, we train another generator to generate the residual signal (or fine structure) of the decoder output. The decoder output is patched with the residual signal to be the final output. The generator is in turn learned with a discriminator which outputs a scalar to indicate whether the input signal is realistic. The discriminator is further trained with an auxiliary classifier, predicting the speaker for the input signal. This helps the generator to produce signals carrying more characteristics of the target speaker.

2. Proposed approach

Let $x \in \mathcal{X}$ be an acoustic feature sequence where $\mathcal{X}$ is the collection of all such sequences, and $y \in \mathcal{Y}$ be a speaker where $\mathcal{Y}$ is the group of all speakers who produce the sequence collection $\mathcal{X}$. The training set $\mathcal{D} = \{(x_1, y_1) \dots (x_m, y_m)\}$ contains m pairs of $(x_i, y_i) \in (\mathcal{X}, \mathcal{Y})$, where the sequence x_i is produced by speaker y_i. During training, x is a fixed-length segment randomly sampled from $\mathcal{X}$. During testing, x can have variable length because the model here is built with recurrent-based components. The whole framework includes two stages of training as shown in Fig. 1 and explained below.

2.1. Stage 1: Autoencoder plus classifier-1

This stage is to learn an autoencoder plus classifier-1.

---

[1]The $\oplus$ in Fig. 1 indicates element-wise addition.
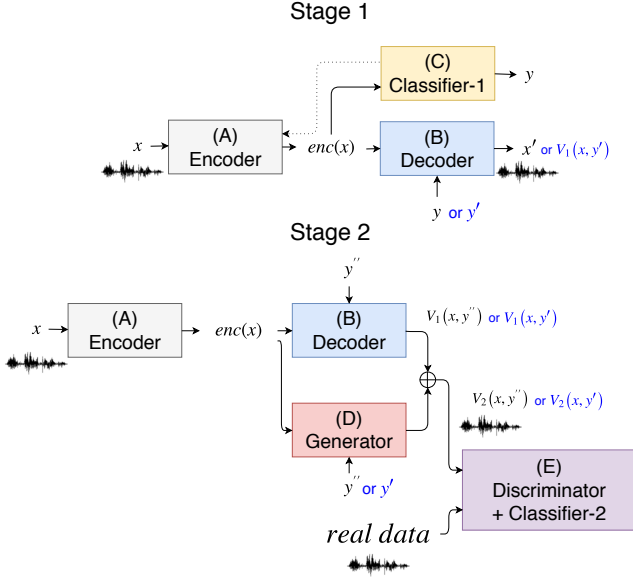
Stage 1

Stage 2

Figure 1: *The training procedure. In stage 1, the encoder (Block(A)) is trained to generate speaker-independent representation with the regularization of the classifier-1 (Block(C)). The decoder (Block(B)) is trained to reconstruct the original acoustic features. In stage 2, a generator (Block(D)) is trained to enhance the output of the decoder (Block(B)) with the help of the Discriminator plus Classifier-2 (Block(E)).*

Autoencoder:

The encoder (Block(A)) is trained to map an input sequence x to a latent representation $enc(x)$. The decoder (Block(B)) is trained to generate x' which is a reconstruction of x from $enc(x)$ given the speaker identity y.

$$x' = dec(enc(x), y). \tag{1}$$

The Mean Absolute Error (MAE) is minimized in training the autoencoder because this generates sharper output than mean square error [29]. So the reconstruction loss is given as in (2)

$$L_{rec}(\theta_{enc}, \theta_{dec}) = \sum_{(x,y)\in\mathcal{D}} \|x' - x\|_1, \tag{2}$$

where θ_{enc} and θ_{dec} are the parameters of the encoder and decoder respectively. This autoencoder alone can achieve VC as below. Given an utterance x produced by a source speaker y, the decoder can generate the voice of a target speaker y' using the linguistic content of x,

$$V_1(x, y') = dec(enc(x), y'). \tag{3}$$

During training, the decoder input is $enc(x)$ and y, but during voice conversion, we replace y with the target speaker y'. $V_1(x, y')$ is the output of stage 1, which has the linguistic content of x, but the identity of y'.

Classifier-1: The autoencoder itself learned with (2) cannot make the latent representation $enc(x)$ speaker-independent. The speaker characteristics of the original speaker x existing in $enc(x)$ inevitably degrades the performance of VC. This is why we train classifier-1 (Block(C)) in addition to regularize the autoencoder to make $enc(x)$ speaker-independent. For each training pair $(x_i, y_i) \in \mathcal{D}$, the classifier takes $enc(x_i)$ as input and outputs the probability $P_{cls1}(y| enc(x_i)), y \in \mathcal{Y}$, which is the probability that x_i is produced by speaker y. The classifier-1 is trained to

minimize the negative log-probability to differentiate the different speakers, as in (4),

$$L_{cls1}(\theta_{enc}, \theta_{cls1}) = \sum_{(x_i,y_i)\in\mathcal{D}} -\log P_{cls1}(y_i| enc(x_i)). \tag{4}$$

On the other hand, however, the encoder is trained to maximize (4) in order to remove the speaker identity in $enc(x)$. So the full objective for the autoencoder regularized by clasifier-1 is

$$L_{ae}(\theta_{enc}, \theta_{dec}, \theta_{cls1}) = L_{rec}(\theta_{enc}, \theta_{dec}) - \lambda L_{cls1}(\theta_{enc}, \theta_{cls1}), \tag{5}$$

which integrates (2) and (4) and λ is a hyper-parameter. The autoencoder and the classifier are trained alternatively.

2.2. Stage 2: GAN

If we simply perform VC with stage 1 as mentioned above, even with the help of classifier-1, the reconstruction loss tends to generate blurry spectra and artifact. This is why in stage 2, we train another pair of generator and discriminator to guide the output spectra to be more realistic [29].

Based on the concept of decoupled learning [30], which stabilize GAN training by decoupling decoder and generator, we separately train another generator (Block(D)) taking $enc(x)$ and the speaker identity y'', which is a uniformly sampled speaker out of all speakers in $\mathcal{Y}$, as the input and generate the residual (or fine structure of the signal) of the output of the decoder (Block(B)). The parameters of the encoder (Block(A)) and the decoder (Block(B)) are fixed in this training stage, which stabilizes the training procedure.

As shown in Fig. 1, the generator here is trained with the help of a "discriminator plus classifier-2" (Block(E)), and the final output during VC test for a selected target speaker $y' \in \mathcal{Y}$ is the addition of the output of the decoder (Block(B)) and the generator (Block(D)), or $y'' = y'$ in Fig. 1,

$$V_2(x, y') = V_1(x, y') + gen(enc(x), y'). \tag{6}$$

In (6), x is the input speech, $enc(x)$ the encoder output, y' the selected target speaker, $V_1(x, y')$ is the converted voice obtained in the stage 1, $gen(enc(x), y')$ the output of the generator, and $V_2(x, y')$ is the VC result for stage 2.

The generator is learned with a discriminator (in Block(E)) in an adversarial network. This discriminator is trained to distinguish whether an input acoustic feature sequence, x, is real or generated by machine. The output of the discriminator $D(x)$ is a scalar indicating how real x is, the larger $D(x)$, the more possible x is real. This adversarial network is trained with the loss in (7), where θ_{gen} and θ_{dis} are the parameters for the generator and the discriminator.

$$L_{adv}(\theta_{gen}, \theta_{dis}) = \sum_{x\in\mathcal{D}} \log(D(x)) + \sum_{x\in\mathcal{D}, y''\sim\text{UNIFORM}(\mathcal{Y})} \log(1 - D(V_2(x, y''))). \tag{7}$$

The discriminator gives larger values to real speech x from the dataset $\mathcal{D}$ in the first term on the right of (7), while assigns lower score to the converted speech $V_2(x, y'')$ in the second term, where y'' is a speaker sampled uniformly from $\mathcal{Y}$. So the discriminator is trained to distinguish real voice and the generated data by maximizing L_{adv} in (7), while on the other hand the generator is trained to fool the discriminator by minimizing L_{adv} in (7).

In addition, the "discriminator plus classifier-2" also includes a classifier-2 which learns to predict the speaker for the speech x

by generating a distribution of speakers $P_{cls2}(y|x)$ [31] based on the training data $\mathcal{D}$. This classifier-2 is trained by minimizing the loss in (8),

$$L_{cls2}^d(\theta_{dis}) = \sum_{(x_i, y_i) \in \mathcal{D}} -\log P_{cls2}(y_i|x_i). \quad (8)$$

This classifier-2 and the discriminator share all layers except with separated last layer. On the other hand, the generator should learn to generate the voice $V_2(x, y'')$ for a uniformly sampled speaker $y'' \in \mathcal{Y}$ which can be predicted as the voice of y'' by the classifier-2, which implies $V_2(x, y'')$ preserves more speaker characteristics of y''. So the generator should be trained to minimize the loss in (9),

$$L_{cls2}^g(\theta_{gen}) = \sum_{x \in \mathcal{D}, y'' \sim \text{UNIFORM}(\mathcal{Y})} -\log P_{cls2}(y''|V_2(x, y'')). \quad (9)$$

Here (8) and (9) are exactly the same, except (8) is for real data and correct speakers in the data set, while (9) for generated voice and sampled target speakers.

So the complete loss function in stage 2 is given as (10) and (11) respectively for the generator θ_{gen} and the discriminator θ_{dis}, trained alternatively, where the first terms on the right of (10) (11) are in (7), and the second terms of (10) (11) are in (8) (9).

$$L_{gen}(\theta_{gen}) = L_{adv}(\theta_{gen}, \theta_{dis}) + L_{cls2}^g(\theta_{gen}) \quad (10)$$

$$L_{dis}(\theta_{dis}) = -L_{adv}(\theta_{gen}, \theta_{dis}) + L_{cls2}^d(\theta_{dis}) \quad (11)$$

3. Implementation

We adopted the model architecture from CBHG module [33]. The detailed network architecture is listed in Table 1. We did not use fully-connected layer across time-steps in order to deal with variable-length input. The convolution-bank aimed to capture local information about the acoustic features. We used the pixel shuffle layer to generate higher resolution spectra [34]. $\text{emb}_l(y)$ indicates the speaker embedding in the l-th layer since the network may need different information in each layer. We plugged the embedding by adding it on the feature map. We used 1d convolution for every network except for the discriminator, which was built with 2d convolution to better capture the texture.

Dropout: We provided the required noise in training with dropout in encoder as suggested [29]. We found it useful to add dropout in the classifier to improve the robustness of the model. We use 0.5 dropout rate in the encoder, and 0.3 in the classifier.

WGAN-GP: GAN is notoriously hard to train. So we applied a different objective function, Wasserstein GAN with gradient penalty (WGAN-GP), to stabilize the training process of GAN [35].

Hyper-parameters: In training stage 1, if we add the classification loss L_{cls1} in (5) at the beginning of the training process, the autoencoder will have problems to reconstruct the spectra well. So we linearly increase the hyper-parameter λ from 0 to 0.01 in the first 50000 iterations to make sure the latent representation became speaker-independent gradually.

4. Experiments

We evaluated our VC model on CSTR VCTK Corpus [36]. The audio data were produced by 109 speakers in English with different

Table 1: *Network architectures. C indicates convolution layer. FC indicates fully-connected layer. Conv1d-bank-K indicates convolution layer with kernel size from 1 to K. LReLU indicates leakyReLU activation. IN indicates instance normalization [32]. Res indicates residual connection. PS indicates pixel shuffle layer for upsampling. The kernel size K for discriminator is 64-128-256-512-512.*

Encoder	
conv-bank block	Conv1d-bank-8, LReLU, IN
conv block × 3	C-512-5, LReLU C-512-5, stride=2, LReLU, IN, Res
dense block × 4	FC-512, IN, Res
recurrent layer	bi-directional GRU-512
combine layer	recurrent output + dense output

Decoder/Generator	
conv block × 3	$\text{emb}_l(y)$, C-1024-3, LReLU, PS C-512-3, LReLU, IN, Res
dense block × 4	$\text{emb}_l(y)$, FC-512, IN, Res
recurrent layer	$\text{emb}_l(y)$, bi-directional GRU-256
combine layer	recurrent output + dense output

Classifier-1	
conv block × 4	C-512-5, LReLU C-512-5, IN, Res
softmax layer	FC-$N_{speaker}$

Discriminator	
conv block × 5	C-K-5, stride=2, LReLU, IN
conv layer	C-32-1
output layer	scalar output, FC-$N_{speaker}$(classifier-2)

accents, such as English, American, and India. Each speaker uttered different sets of sentences. We selected a subset of 20 speakers, 10 females and 10 males, as $\mathcal{Y}$ mentioned above. The dataset was randomly split to training and testing sets by the percentage 90% and 10%.

We used log-magnitude spectrogram as the acoustic features. The detailed spectral analysis and synthesis setting was the same as the previous work [33].

Training details: We trained the network using Adam optimizer with learning rate $lr = 0.0001$, $\beta_1 = 0.5$, $\beta_2 = 0.9$. Batch size was 32. We randomly sampled 128 frames of spectrogram with overlap. We trained classifier/discriminator for 5 iterations and 1 iteration for encoder/generator.

We first pretrained the encoder and decoder with L_{rec} in (2) for 8000 mini-batches, then pretrained the classifier-1 in with L_{cls1} in (4) for 20000 mini-batches. Stage 1 was trained for 80000 mini-batches, and stage 2 another 50000 mini-batches.

4.1. Objective evaluation

Diversified distribution over all frequencies is a highly desired property of speech signals, and the over-smoothed spectra generated by many conventional approaches has been a major problem of voice conversion [18]. This property can be observed by calculating the Global Variance (GV) over the spectrum. Higher global variance indicates sharpness of the converted speech. We evaluated the global variance for each of the frequency index for 4 conversion examples: male to male, male to female, female to
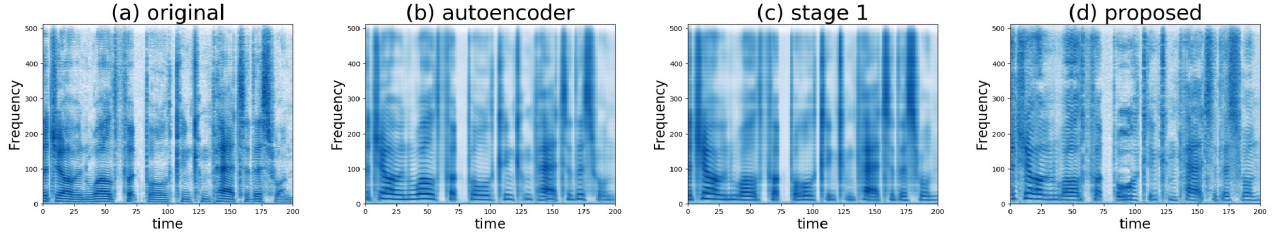
Figure 2: *The heatmaps of the spectrogram: (a) original voice, (b)(c)(d) converted voice, (b) autoencoder of stage 1 alone without classifier-1, (c) complete Stage 1 with classifier-1, and (d) proposed approach incluiding Stage 1 and 2.*
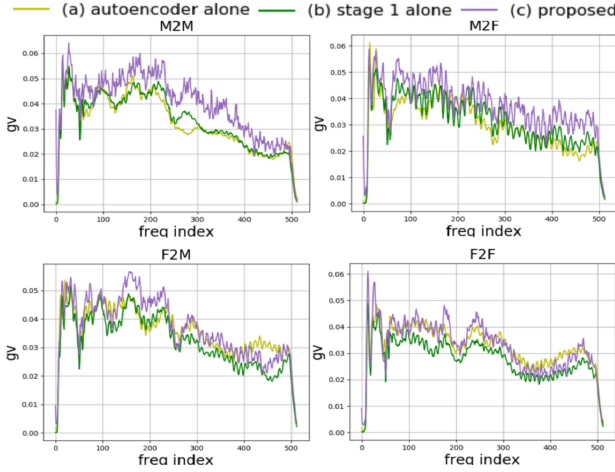

Figure 3: *The global variance for each frequency index of spectrogram for 4 conversion examples: M2M, M2F, F2M, F2F, where F indicates female speaker and M indicates male speaker.*

Table 2: *The averages of the global variance for 4 conversion examples: M2M, M2F, F2M, F2F, where F indicates female speaker and M indicates male.*

	M2M	M2F	F2M	F2F
(a) autoencoder alone	0.0340	0.0334	0.0341	0.0307
(b) stage 1 alone	0.0326	0.0338	0.0322	0.0272
(c) proposed	0.0394	0.0401	0.0389	0.0333

male, and female to female. The results are in Fig. 3. In each example 3 curves for 3 cases are plotted: (a) autoencoder of stage 1 alone without classifier-1, (b) complete stage 1 with classifier-1 and (c) proposed approach with stage 1 and stage 2. We can see in all cases the proposed approach (curves (c)) offered the best sharpness. Averages over all frequencies for those curves in Fig. 3 are listed in Table 2, from which it is clear the proposed approach (row(c)) offered the highest global variance. A set of example spectrogram is in Fig. 2 for the original voice (a) and converted (b) (c) (d), where the sharpness offered by the proposed approach can be observed.

4.2. Subjective evaluation

We also performed subjective human evaluation for the converted voice. 20 subjects were given pairs of converted voice in random order and asked which one they preferred in terms of two measures: the naturalness and the similarity in speaker characteristics to a referenced target utterance produced by the target speaker.
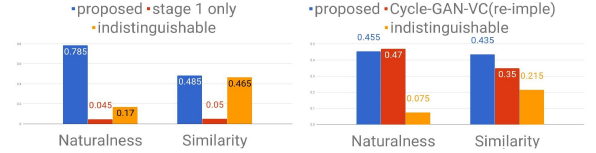

Figure 4: *Results of subjective preference test in naturalness and similarity in speaker characteristics. The left compared the proposed approach with stage 1 only. The right compared the proposed approach with Cycle-GAN-VC [25].*

Average of examples including intra-gender and inter-gender conversion are shown in Fig. 4. The ablation experiment on the left of Fig. 4 compared two method: the proposed approach including stages 1 and 2, and complete stage 1 with classfier-1 but not stage 2. We can see stage 2 has significantly improved the voice quality in terms of both naturalness and similarity in speaker characteristics. Here we also compared the proposed approach with a re-implementation of Cycle-GAN-VC [25], a previous work comparable to methods utilizing parallel data. The result is on the right of Fig. 4. As the result shows, the proposed approach is comparable in terms of the naturalness and the similarity in speaker characteristics, while considering to multi-speakers without parallel data. [2]

4.3. Degree of disentanglement

To evaluate the degree of disentanglement of our model with respect to speaker characteristics, we trained another speaker verification network that takes the latent representation $enc(x)$ as input to predict the speaker identity [37]. The speaker verification network has the same architecture as the classifier-1 in stage 1. The verification accuracy was 0.916 without the classifier-1, but dropped significantly to 0.451 when classifier-1 was added. This verified that the classifier-1 successfully disentangled the speaker characteristics from the latent representation.

5. Conclusion

We proposed an approach for voice conversion by extracting the speaker-independent representation. No parallel data are needed and conversion to multiple target speakers can be achieved by a single model. We show that adding a residual signal can improve significantly the quality of converted speech. Objective evaluation metrics of global variance show that sharp voice spectra can be produced with this approach. This is also verified with subjective human evaluation.

---

[2]Demo webpage: `https://jjery2243542.github.io/voice_conversion_demo/`

6. References

[1] 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.

[2] A. Kain and M. W. Macon, "Spectral voice conversion for text-to-speech synthesis," in *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, vol. 1. IEEE, 1998, pp. 285–288.

[3] H. Miyoshi, Y. Saito, S. Takamichi, and H. Saruwatari, "Voice conversion using sequence-to-sequence learning of context posterior probabilities," *arXiv preprint arXiv:1704.02360*, 2017.

[4] T. Nakashika, T. Takiguchi, and Y. Ariki, "Voice conversion based on speaker-dependent restricted boltzmann machines," *IEICE TRANSACTIONS on Information and Systems*, vol. 97, no. 6, pp. 1403–1410, 2014.

[5] D. Saito, K. Yamamoto, N. Minematsu, and K. Hirose, "One-to-many voice conversion based on tensor representation of speaker space," in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

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

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

[8] T. Toda, M. Nakagiri, and K. Shikano, "Statistical voice conversion techniques for body-conducted unvoiced speech enhancement," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 9, pp. 2505–2517, 2012.

[9] S. A. Kumar and C. S. Kumar, "Improving the intelligibility of dysarthric speech towards enhancing the effectiveness of speech therapy," in *Advances in Computing, Communications and Informatics (ICACCI), 2016 International Conference on*. IEEE, 2016, pp. 1000–1005.

[10] Z. Inanoglu and S. Young, "Data-driven emotion conversion in spoken english," *Speech Communication*, vol. 51, no. 3, pp. 268–283, 2009.

[11] 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.

[12] S. H. Mohammadi and A. Kain, "Voice conversion using deep neural networks with speaker-independent pre-training," in *Spoken Language Technology Workshop (SLT), 2014 IEEE*. IEEE, 2014, pp. 19–23.

[13] T. Nakashika, T. Takiguchi, and Y. Ariki, "High-order sequence modeling using speaker-dependent recurrent temporal restricted boltzmann machines for voice conversion," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[14] L. Sun, S. Kang, K. Li, and H. Meng, "Voice conversion using deep bidirectional long short-term memory based recurrent neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4869–4873.

[15] L.-H. Chen, Z.-H. Ling, L.-J. Liu, and L.-R. Dai, "Voice conversion using deep neural networks with layer-wise generative training," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 22, no. 12, pp. 1859–1872, 2014.

[16] S. Takamichi, T. Toda, G. Neubig, S. Sakti, and S. Nakamura, "A postfilter to modify the modulation spectrum in hmm-based speech synthesis," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 290–294.

[17] Y. Ohtani, T. Toda, H. Saruwatari, and K. Shikano, "Maximum likelihood voice conversion based on gmm with straight mixed excitation," 2006.

[18] T. Toda, A. W. Black, and K. Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 8, pp. 2222–2235, 2007.

[19] E. Helander, T. Virtanen, J. Nurminen, and M. Gabbouj, "Voice conversion using partial least squares regression," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 5, pp. 912–921, 2010.

[20] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[21] 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.

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

[23] ——, "Voice conversion from unaligned corpora using variational autoencoding wasserstein generative adversarial networks," *arXiv preprint arXiv:1704.00849*, 2017.

[24] W.-N. Hsu, Y. Zhang, and J. Glass, "Unsupervised learning of disentangled and interpretable representations from sequential data," in *Advances in neural information processing systems*, 2017, pp. 1876–1887.

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

[26] Y. Gao, R. Singh, and B. Raj, "Voice impersonation using generative adversarial networks," *arXiv preprint arXiv:1802.06840*, 2018.

[27] G. Lample, N. Zeghidour, N. Usunier, A. Bordes, L. Denoyer *et al.*, "Fader networks: Manipulating images by sliding attributes," in *Advances in Neural Information Processing Systems*, 2017, pp. 5969–5978.

[28] 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," *arXiv preprint arXiv:1711.09020*, 2017.

[29] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *arXiv preprint*, 2017.

[30] Z. Zhang, Y. Song, and H. Qi, "Decoupled learning for conditional adversarial networks," *arXiv preprint arXiv:1801.06790*, 2018.

[31] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier gans," *arXiv preprint arXiv:1610.09585*, 2016.

[32] D. Ulyanov, A. Vedaldi, and V. S. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *CoRR*, vol. abs/1607.08022, 2016. [Online]. Available: http://arxiv.org/abs/1607.08022

[33] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio *et al.*, "Tacotron: Towards end-to-end speech syn," *arXiv preprint arXiv:1703.10135*, 2017.

[34] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1874–1883.

[35] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in Neural Information Processing Systems*, 2017, pp. 5769–5779.

[36] C. Veaux, J. Yamagishi, K. MacDonald *et al.*, "Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit," 2017.

[37] E. L. Denton *et al.*, "Unsupervised learning of disentangled representations from video," in *Advances in Neural Information Processing Systems*, 2017, pp. 4417–4426.