Non-parallel Many-to-many Singing Voice Conversion by Adversarial Learning

Jinsen Hu, Chunyan Yu*, Faqian Guan

College of Mathematics and Computer Science, Fuzhou University, China
therica@fzu.edu.cn, {hhuu1991, faqianguan}@gmail.com

Abstract*— With the rapid development of deep learning, although speech conversion had made great progress, there are still rare researches in deep learning to model on singing voice conversion, which is mainly based on statistical methods at present and can only achieve one-to-one conversion with parallel training datasets. So far, its application is limited. This paper proposes a generative adversarial learning model, MSVC-GAN, for many-to-many singing voice conversion using non-parallel datasets. First, the generator of our model is concatenated by the singer label, which denotes domain constraint. Furthermore, the model integrates self-attention mechanism to capture long-term dependence on the spectral features. Finally, switchable normalization is employed to stabilize network training. Both the objective and subjective evaluation results show that our model achieves the highest similarity and naturalness not only on the parallel speech dataset but also on the non-parallel singing dataset.

I. INTRODUCTION

Singing is the musical sound produced by humans through the vocal organs. Compared with the music produced by the instrument, in addition to the changes in pitch, rhythm, and melody, singers' timbre brings extra expressive power. Timbre is an important sensory feature that distinguishes sound from others, with the same loudness and pitch. The human voice timbre is different due to the vibration of the vocal cords, the air supply when the sound is pronounced, and the shape and size of the vocal tract. A singer can change his timbre through singing skills in a certain range. But this change is biologically constrained by his vocal organs.

The singing voice conversion transforms singing with the timbre of a source singer into singing with a different timbre of a target singer, with the lyrics and pitch remaining unchanged as if it was sung by the target singer [1,2].

At present, it is lack of research on singing voice conversion and is common to convert singing voice using those methods from speech conversion, such as Gaussian mixture model (GMM) [3,4], eigenvoice based conversion [5,6], exemplar-based method [7,8], and neural network methods including long-short-term memory network [9,10] and convolutional neural networks (CNN)[11,12,13]. Most methods are one-to-one and parallel. A one-to-one conversion model means that it can only convert the timbre of a specific source into that of a specific target. A parallel conversion model means that it is trained by parallel datasets in which the audio of the source and target has exactly the same semantic content.

GMM [3] is the most widely used speech conversion method. It can model the joint probability distribution of the source and target domain, and construct an accurate and continuous model based on fewer data. However, the formant characteristics became unobvious after smoothing processing on the spectral features. Based on the spectral parameter trajectory, [4] proposed a maximum likelihood estimation conversion method to overcome over-smoothing but lead to over-fitting.

Similarly, based on these statistical models, most singing voice conversion methods employ small parallel datasets to train parametric models. [1] was the first to apply the GMM to the singing voice conversion, considering the difference between singing and speech, that is, the pitch of the singing voice is limited to a fixed range. [6] applied the eigenvoice conversion technique to the singing voice, and realized conversion between arbitrary singers through a combination of one-to-many and many-to-one conversion models. However, parallel datasets were still needed in the training stage.

Recently, some researchers have employed deep neural networks for speech conversion to overcome the shortcomings of traditional models that can only use parallel datasets. [11,12,13] presented some deep generative models for non-parallel and many-to-many conversion. [14] employed two generators to learn both forward and reverse mapping between the source and target speakers' acoustic features, respectively, and to perform the one-to-one conversion. Although deep generative models for speech conversion has been exploited, for singing voice conversion, it has not been reported.

For singing voice conversion models [15,16], parallel and one-to-one limited their applications. On one hand, one-to-one leads to poor generalization ability. On the other hand, parallel song datasets are very difficult to collect. Furthermore, even if a parallel dataset is collected, further careful alignment is required, which is also difficult and time-consuming.

Hence, it is necessary to employing a data-driven method to seek a non-parallel and many-to-many singing voice conversion model. The model should directly learn the characteristics of the vocal spectral structure and no further spectral alignment is needed.

Inspired by [14,17,18], we present a singing voice conversion model, MSVC-GAN, which means many-to-many singing voice conversion using GAN [19] based method. It is a cycle-consistent adversarial network integrated with a self-attention mechanism [20] and switchable normalization (SN) [21]. Through adversarial loss, MSVC-GAN learns the distribution of acoustic characteristics of different singers, and establishes forward and reverse mapping among different singer acoustic features. It means that the model can learn a sole function for all many-to-many mapping. Furthermore, through the self-attention mechanism and reconstruction loss, MSVC-GAN learns the details of the spectral features, which leads to change timbre with semantic content remaining unchanged. According to our understanding, this paper is the first study on the generative adversarial network for many-to-many singing voice conversion under non-parallel datasets.

II. MSVC-GAN: A NON-PARALLEL AND MANY-TO-MANY SINGING VOICE CONVERSION MODEL

A. Fundamental of StarGAN

StarGAN [17] was originally applied to multi-domain image-to-image translation. Unlike CycleGAN [18], StarGAN learns characteristics of each domain with one generator and one discriminator to establish mappings among multiple domains with non-parallel datasets.

To learn the features among multiple domains at the same time, the domain label c is introduced as a constraint on the generator G during training. That is, for an input image x under the constraint of the target domain c, the output image $xout = G(x, c)$. If the target domain c is randomly changed, it can make G adapt to the transformation between multiple domains. Meanwhile, to enable D to adapt to multiple domains, an auxiliary classifier C is appended, to help D learn domain features and discriminate real or fake samples simultaneously.

B. Architecture of MSVC-GAN

Similar to multi-domain image-to-image translation, in singing voice conversion, different singers are regarded as different domains. Inspired by [14,17,18], we propose a non-parallel and many-to-many singing voice conversion model, MSVC-GAN. The architecture is shown in Fig. 1. The generator G is concatenated by singer ID, referred to as c in the paper, corresponding to domain information.

We consider singing voice conversion as a conversion of spectral features. The spectral structure is closely related to time and frequency. To better learn the time-frequency characteristics of spectral, we use a 2D convolution network to learn the spectral time-frequency features simultaneously. Especially, convolution kernels are of different shapes. This enables us to learn more effective spectral features [22,23].

The discriminator D is a Patch GAN discriminator [24]. It uses Leaky ReLU nonlinearity and employs no normalization. D solves a binary classification task determine whether input spectrograms are real spectrograms of source singer or translation output coming from G. The domain classifier C consists of several convolutional layers followed by BN, ReLU nonlinearity, and MaxPool. The nonlinearity and normalization operations included in the network are excluded in the visualization for avoiding a cluttered presentation. D and C are fully convolutional which allow input of any length.

The generator adopts the encoder-decoder structure. The encoder consists of several 2D convolution layers followed by several residual blocks [25]. It maps the input Mel-cepstral coefficients (MCC) to a latent code which is a spatial feature map. The decoder is made of several residual blocks followed by a few transpose convolutional layers. We use SN to normalize the feature maps in the encoder and decoder. The speaker ID is concatenated to the decoder. This gives the decoder a constraint to output the desired spectral in the target domain. Spectral has an obvious hierarchical structure in time and frequency. CNN can learn the local pattern of features, but ignore the consistency of global patterns of features. Due to the limitation of the size of the convolution kernel, similarly, the convolution operation in GAN can only learn the local structural features of the spectral, and it is difficult to capture the long-term dependence of the spectral. Hence, to better reconstruct the spectral details, several attention layers are inserted in the decoder.

The self-attention mechanism [18] is formalized as

$$\beta_{j,i} = \frac{\exp\left(f(x_i)^T \cdot g(x_j)\right)}{\sum_{i=1}^{N} \exp\left(f(x_i)^T \cdot g(x_j)\right)} \tag{1}$$

The spectral features from the previously hidden layer $x \in \mathbb{R}^{C \times N}$ are first transformed into two feature spaces f, g to calculate the attention, where $f(\mathrm{x}) = \mathrm{W}_f x$, $g(\mathrm{x}) = \mathrm{W}_g x$ .and $\beta_{j,i}$ is a two-dimensional attention matrix, indicating the extent to which the model attends to the i^{th} location when synthesizing the j^{th} region.

The output of the attention layer is $\mathrm{o} = \left(o_1, o_2, \ldots, o_j, \ldots, o_N\right) \in \mathbb{R}^{C \times N}$, where,

$$o_j = \sum_{i=1}^{N} \beta_{j,i} h(x_i) \tag{2}$$

In the above formulation, $W_f \in \mathbb{R}^{\bar{C} \times C}$, $W_g \in \mathbb{R}^{\bar{C} \times C}$, $W_h \in \mathbb{R}^{C \times C}$ have learned weight matrices during training, which are implemented by 1×1 convolutions. We use $\bar{C} = C/8$ in all the experiments.

The final output is obtained through multiplying the output of the attention layer by the scale parameter and adding to the original feature map. The formula is

$$z_i = \gamma o_i + x_i \tag{3}$$

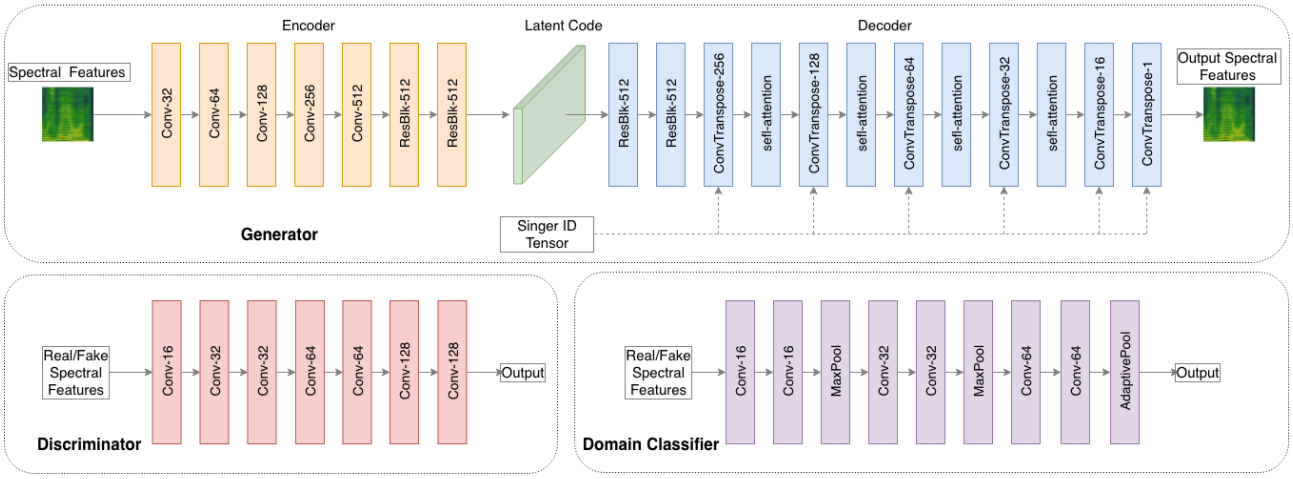

Fig. 1 The architecture of the proposed MSVC-GAN model. The number in each block denotes the number of filters in the layer. The generator input is 80 Mel-cepstral coefficients with 256 frames. The singer ID tensor is obtained by tile the one-hot representation of singer ID and then concatenated to feature maps coming from transpose convolution layers. Real spectral features mean spectral features coming from the training set while Fake spectral features coming from generator output.

γ is initialized to 0 and learned during training. This allows the model to gradually learn the attention matrix during training.

Normalization is an important component in the GAN-based model as it can stable GAN training. Different normalizers, like Batch Normalization (BN) [26], Instance Normalization (IN) [27] and Layer Normalization (LN) [28], are often used to solve different tasks. And, existing practices often employed the same normalizer in all normalization layers of an entire network, rendering suboptimal performance. Switchable Normalization (SN) [21] combines three types of statistics estimated channel-wise, layer-wise, and minibatch-wise by using IN, LN, and BN respectively. We use SN in the MSVC-GAN model, as SN combines the advantages of the three normalization methods and automatically selects the appropriate normalization method based on the training objectives. Experiments verified the effectiveness of SN.

C. Loss in MSVC-GAN

MSVC-GAN involves adversarial loss, domain classification loss, reconstruction loss, and identify-mapping loss.

Adversarial loss L_{adv} measures how much of the generated sample is like a real sample of the target domain. Optimizing it helps G generates more realistic samples. It is defined as

$$L_{adv} = \mathbb{E}_x[logD_{src}(x)] + \mathbb{E}_{x,c}\left[\log\left(1 - D_{src}\left(G(x,c)\right)\right)\right] \quad (4)$$

The goal of G is that the generated sample $G(x,c)$ under constraint c should be as close as possible to real sample in the target domain, and the goal of the D is to judge whether the input sample is generated or a real one in the target domain. D_{src} refers to the probability distribution over sources given by D. During training, G minimizes L_{adv} while D maximizes it.

For G, given input sample x and the target domain label c, the output y should be able to be classified as target domain c. The classification loss should optimize both G and D. That is, the classification loss of real samples is used to optimize D while that of fake samples to optimize G. Both are defined as

$$L_{cls}^{r} = \mathbb{E}_{x,c'}[-logC_{cls}(c'|x)] \quad (5)$$

$$L_{cls}^{f} = \mathbb{E}_{x,c}\left[-logC_{cls}\left(c\middle|G(x,c)\right)\right] \quad (6)$$

Where $C_{cls}(c'|x)$ represents a probability distribution over domain labels computed by C. Minimizing L_{cls}^{r}, C learns to classify the real sample x into its original domain c'. Similarly, minimizing L_{cls}^{f}, G tries to generate samples that can be classified into target domain c.

Although minimizing L_{adv} enables G to learn the data distribution of the target domain and generate samples more liking those real samples in the target domain, this cannot guarantee that the content of the input sample x will be preserved. Hence, we introduced reconstruction loss, formulated as:

$$L_{rec} = \mathbb{E}_{x,c,c'}[\|x - G(G(x,c),c')\|_1] \quad (7)$$

$G\left(G(x,c),c'\right)$ denotes that $G(x,c)$ is input to G again to generate a sample under the constraint of source domain label c'.

Although L_{rec} is effective for reconstructing the structure of the spectral, there is no guarantee that the semantic content of the input will always be preserved. Hence, we introduce identity-mapping loss, formulated as

$$L_{id} = \mathbb{E}_{x,c}[\|G(x,c') - x\|_1] \quad (8)$$

$G(x,c')$ denotes that sample x from the source domain is input to G to generate another sample in the source domain. By calculating the L1 loss for x and $G(x,c')$,

L_{id} makes G trying to maintain the consistency of the source domain features.

The complete loss functions to optimize G and D are written, respectively, as

$$L_D = -L_{adv} + \lambda_{cls} L^r_{cls} \quad (9)$$

$$L_G = L_{adv} + \lambda_{cls} L^f_{cls} + \lambda_{rec} L_{rec} + \lambda_{id} L_{id} \quad (10)$$

Where λ_{rec}, λ_{cls} and λ_{id} are hyperparameters that control the relative importance of reconstruction, domain classification and identity mapping losses, respectively, compared to the adversarial loss. We use $\lambda_{rec} = 10, \lambda_{cls} = 1$ and $\lambda_{id} = 3$ in all of our experiments.

III. EXPERIMENTS

A. Datasets and experiment setting

To evaluate the performance of MSVC-GAN, experiments are performed on parallel speech and non-parallel singing datasets.

The parallel speech dataset is VCC2016 [29]. Like [14], we select two males (TM1, TM2), and two females (SF1, SF2) as our training set, denoted as S*# in the following. * is F for female or M for male, and # is the serial number. The audio files for each speaker were manually segmented into 216 short parallel sentences (about 13 minutes). Among them, 162 and 54 sentences were provided as training and testing sets, respectively. Although it is parallel, no alignment is performed when training MSVC-GAN.

There are no public singing datasets for the reason of copyright. Hence, we collect a lot of free Chinese popular songs from the Internet and use the vocal separation algorithm [30] to extract the vocals. Then, we select 4 singers, 2 males, and 2 females, with 7 to 9 songs of each singer. Moreover, after removing the parts that do not contain vocal, we divide the all singing voice into segments of 3-4s. Finally, for each singer, we randomly choose 290 segments. Among them, 240 (about 14 minutes) and 50 segments are used as training and testing sets, respectively.

In the data preprocessing process, all speech and singing are resampled to 16kHZ with 16bit bit-depth. The WORLD analysis system [31] is used to extract 80 Mel-cepstral coefficients (MCC), logarithmic fundamental frequency ($\log F_0$) and aperiodicity (AP) every 5 ms. Among these features, the model learned a mapping in the MCC domain. Source singer's $\log F_0$ is linearly transformed by equalizing the mean and standard deviation of the target singer's $\log F_0$, and AP keeps unchanged as it has no significant impact on the quality of synthesized speech [11,12].

We conduct both objective and subjective evaluations. The objective evaluation metric is speaker/singer-identity (SD), which is the probability that a segment belongs to a target speaker/singer. And the subjective evaluations assess the naturalness of the converted song.

Convolution operation often used in deep-learning-based on speaker recognition algorithms that seek to determine the identity of a speaker from audio [32,33]. Thus, in objective evaluations, we employ a CNN-based classifier, including 5 convolution layers and 2 fully connected layers, for speaker/singer recognition. The architecture is illustrated via the following chain of operations: Conv-16 → Conv-32 → Conv-64 → MaxPool3 × 3→ Conv-128 → MaxPool2 × 2 → Conv-256 → MaxPool2 × 2 → Dense-128 →Dropout → Dense-||S|| where ||S|| is the number of speakers or singers. Mel-scale spectrograms are extracted from speech and singing to train the classifier, respectively. First, we perform STFT to the audio with a 60 ms window, a 20 ms hop length. And then, the magnitude spectrograms are transformed into 64-bin Mel-scale spectrograms. Tables 1 and 2 depict the recognition accuracy of our CNN-based classifier on the speech and singing voice test sets, respectively.

Table 1: Performance of CNN-based classifier on speaker recognition

SF1	SF2	SM1	SM2
0.97	0.94	0.94	0.96

Table 2: Performance of CNN-based classifier on singer recognition

F1	F2	M1	M2
0.80	0.82	0.87	0.86

The CNN-based classifier has achieved a good classification accuracy on VCC2016 and a relatively poor accuracy on the singing voice dataset. This probably because that singing voice is acquired by vocal separation method, contains some noise, and pitch in singing changes more drastically than that in speech.

B. Objective evaluation

GMM [4] is often chosen as a baseline for a parallel speech conversion experiment. For non-parallel one-to-one speech conversion, CycleGAN-VC [14] has achieved the best results that are comparable with GMM. Hence, we choose GMM and CycleGAN-VC as two comparison methods. Furthermore, to verify the effectiveness of self-attention mechanism and SN in MSVC-GAN, the experiment also compares the performance of the basic GAN model, basic GAN models with self-attention or SN, denoted as GAN w/attention or GAN w/SN in the following. Note that, MSVC-GAN is a basic GAN model equipped with self-attention and SN.

The converted sing voice and speech are classified by the above CNN-based classifier. Taking the speech conversion as an example, the converted speech is input into the classifier. The classification accuracy of converted speech has two aspects. One is the accuracy of classified as the target, and the other is that of classified as the source. The higher the probability of classified as the target, the more timbre of the target converted. The lower the probability of classified as the source, the better the conversion effect. Hence, the SD of converted speech through classifier can reflect conversion validity.

Four different combinations of parallel speech conversion, female to male (SF1-SM1), male to female (SM1-SF1), female to female (SF2-SF1), male to male (SM2-SM1), are evaluated. In each combination, the former speaker is the source speaker, and the latter is the target speaker Table 3 shows the comparison results.

For non-parallel singing voice conversion, four combinations, female to male (F1-M1), male to female (M1-F1), female to female (F2-F1), male to male (M2-M1), are also evaluated. Without a parallel dataset, GMM is excluded. Table 4 shows the comparison results. When combined, our MSVC-GAN outperforms other comparable models. To be specific, for MSVC-GAN, the probability of classified as the target is the highest, and that of classified as the source is the lowest, except for the speech conversions from male to female and male to male.

Compared to speech conversion, the performance of all GAN-related models for singing voice conversion decreases due to non-parallel and noisy singing datasets. The performance drops of MSVC-GAN are the smallest. This means that our MSVC-GAN has the best generalization ability.

The overall performance of CycleGAN-VC is slightly worse than that of our MSVC-GAN. It is important to note that CycleGAN-VC is a one-to-one model while MSVC-GAN is many-to-many. That is, the CycleGAN-VC model needs to be retrained for a different source or target. Obviously, MSVC-GAN, doing all conversions within one model, achieves a lower calculating cost with fewer parameters and one-time training.

And, the overall performances of GAN w/SN and GAN w/attention are fully beyond that of the basic model and close to that of CycleGAN-VC. This indicates that the self-attention mechanism, maintaining the long-term dependence consistency of the spectral global features, and SN, stabilizing GAN training, are effective for GAN architecture.

Moreover, MSVC-GAN is evaluated for all converting combinations of each speaker/singer pair. There is a total of 16 combinations, respectively. Fig. 2 and Fig. 3 illustrates the confusion matrix. The horizontal axis represents the prediction label and the vertical axis denotes the conversion combination. Without exception, in all combinations, for converted speech and singing voice, the probability classified as the target is significantly greater than that classified as the source.

C. Subjective evaluation

The generator of the MSVC-GAN model extracts and reconstructs spectral features. Fig. 4 illustrates the spectrograms of a converted singing voice for GAN, CycleGAN-VC, and MSVC-GAN. The spectrogram on the upper row is for F1-M1 conversion and that on the lower four pictures for M1-F1 conversion. Compared to the spectrogram for the source on the leftmost column, the spectrogram details generated by MSVC-GAN are clearer, the harmonics are richer, and the energy is stronger. Note that, the leftmost column is singing voice obtained by vocal separation algorithm, thus the spectral structure is a little corrupted.

Furthermore, we also make a subjective evaluation for the similarly and naturalness of a converted singing voice for GAN, CycleGAN-VC, and MSVC-GAN. 20 participants evaluated four combinations of conversion. For each combination, 20 song segments are randomly selected from all converted songs. The listener first evaluates the similarity of the generated voice to the target singing voice, and then evaluate the naturalness of the converted song, both on a scale between 1–5.

Table 3: Comparison of results of converted speech classification accuracy. Taking SF1-SM1 sub-column for example, the number under SF1 means classification accuracy of classified as source speaker SF1, similarly, the number under SM1 means classification accuracy of classified as target speaker.

	Many-to-many	Non-parallel	SF1-SM1		SF2-SF1		SM1-SF1		SM2- SM1	
			SF1	SM1	SF2	SF1	SM1	SF1	SM2	SM1
GMM	×	×	0.06	0.83	0.18	0.81	**0.00**	**0.97**	**0.04**	**0.90**
GAN	√	√	0.03	0.76	0.15	0.80	0.02	0.84	0.20	0.66
GAN w/SN	√	√	0.01	0.78	0.12	0.82	0.02	0.89	0.20	0.66
GAN w/attention	√	√	0.01	0.85	0.15	0.81	0.01	0.89	0.15	0.74
CycleGAN-VC	×	√	0.01	0.77	**0.12**	**0.85**	0.02	0.91	0.09	0.84
MSVC-GAN	√	√	**0.01**	**0.85**	**0.12**	**0.85**	0.01	0.91	0.10	0.86

Table 4: Comparison results of converted songs classification. Taking F1-M1 sub-column for example, the number under F1 means classification accuracy of classified as source speaker F1, similarly, the number under M1 means classification accuracy of classified as target speaker M1.

	F1-M1		F2-F1		M1-F1		M2-M1	
	F1	M1	M1	F1	M1	F1	M2	M1
GAN	0.13	0.57	0.15	0.58	0.10	0.50	0.21	0.55
GAN w/SN	0.12	0.64	0.11	0.65	0.14	0.55	**0.11**	0.69
GAN w/attention	0.09	0.61	0.12	0.60	0.13	0.50	0.24	0.60
CycleGAN-VC	0.07	0.71	0.14	0.59	0.07	0.55	0.16	0.66
MSVC-GAN	**0.04**	**0.71**	**0.08**	**0.69**	**0.05**	**0.69**	0.16	**0.70**

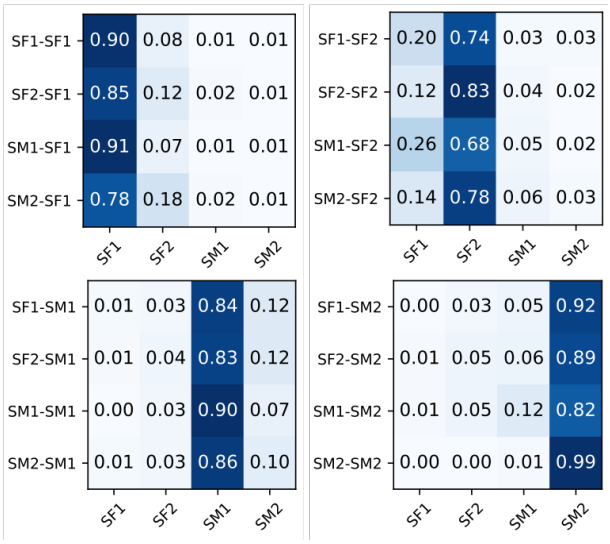

Fig. 2 Confusion matrix of the converted speech.

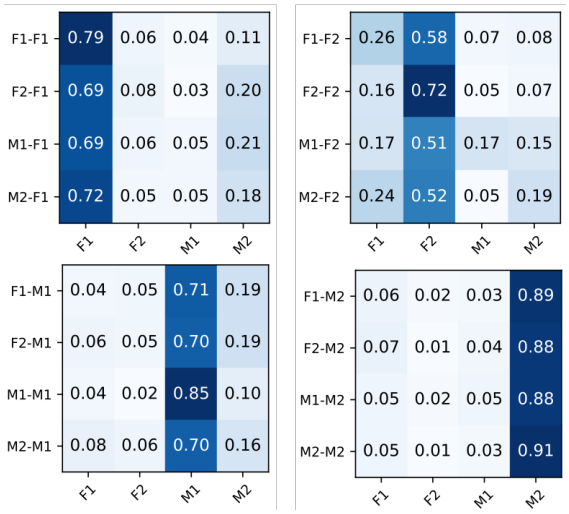

Fig. 3 Confusion matrix of the converted songs.

As shown in Fig.5, MSVC-GAN outperforms the other two models with the highest scores for each combination in terms of similarly and naturalness. GAN has the lowest natural and similarity scores. Especially, although CycleGAN-VC, as a one-to-one model, training a specific conversion model for each converting combination, its natural scores and similarity are still lower than MSVC-GAN, which employs a single conversion model for all combinations. In the experiment, compared to a pure speech, we find CycleGAN-VC sometimes fails to translate the identity of the noisy singing voice, whereas our model integrated with self-attention and SN is more robust to noisy singing voice dataset.

IV. Conclusion

Non-parallel and many-to-many are two big challenges for singing voice conversion. MSVC-GAN is a cycle-consistent generative adversarial learning model integrated with self-attention mechanism and switchable normalization. Different from CycleGAN-VC which is a one-to-one conversion model, MSVC-GAN learns spectral features of each singer with one generator and one discriminator and establishes mappings among multiple singers with non-parallel datasets, through an auxiliary classifier. Experiment results on both parallel speech conversion and non-parallel singing voice conversion show that MSVC-GAN outperforms other conversion models, whether one-to-one or many-to-many, in either objective or subjective evaluations. Experiment results also confirm that the self-attention mechanism and SN can improve the converted audio quality and timbre similarity.

V. Acknowledgment

This work is supported by the Fujian Foundation of Science [grant number 2018J01794] and the Fujian Health-Education Joint Foundation [grant number WKJ2016-6-26].

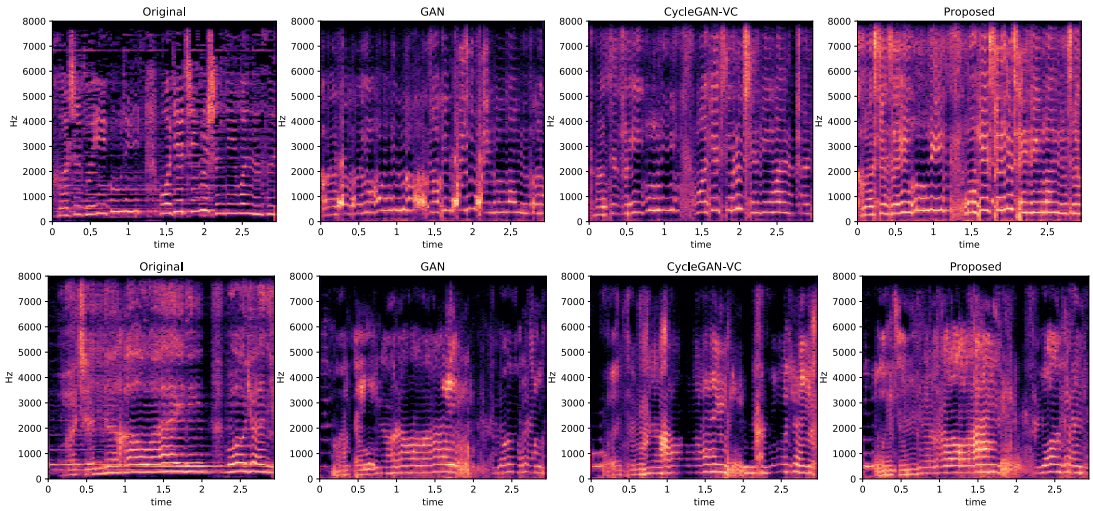

Fig. 4 Spectrogram comparison of GAN-related models.

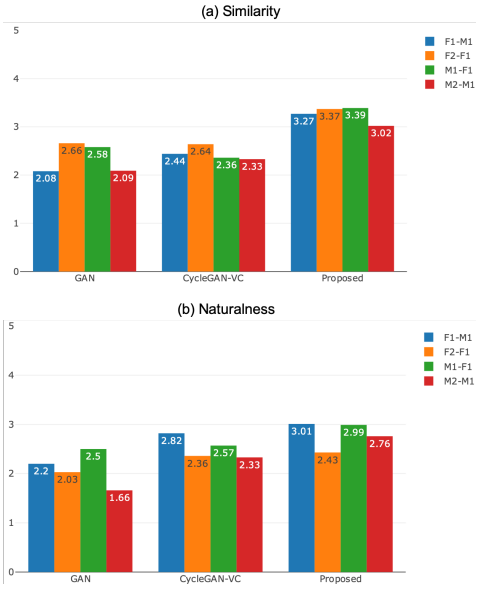

Fig. 5 Similarly and naturalness of converted singing voice with four kinds of conversion.

VI. REFERENCES

[1] F. Villavicencio and J. Bonada, "Applying voice conversion to concatenative singing-voice synthesis," in *Proc. INTERSPEECH*, 2010, pp. 2162-2165.

[2] Y. Kawakami, H. Banno, and F. Itakura, "GMM voice conversion of singing voice using vocal tract area function," *IEICE technical report. Speech 110(297) (Japanese edition)*, pp. 71–76, Nov. 2010.

[3] Y. Stylianou, O. Cappé, and E. Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Trans. Speech and Audio Processing*, vol. 6, no. 2, pp. 131–142, 1998.

[4] T. Toda, A. W. Black, and K. Tokuda, "Voice Conversion Based on Maximum-Likelihood Estimation of Spectral Parameter Trajectory," *IEEE Trans. Audio, Speech & Language Processing*, vol. 15, no. 8, pp. 2222–2235, 2007.

[5] T. Toda, Y. Ohtani, and K. Shikano, "One-to-Many and Many-to-One Voice Conversion Based on Eigenvoices," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2007, pp. IV1249-IV1252.

[6] H. Doi, T. Toda, T. Nakano, M. Goto and S. Nakamura, "Singing voice conversion method based on many-to-many eigenvoice conversion and training data generation using a singing-to-singing synthesis system," *Proceedings of The 2012 Asia Pacific Signal and Information Processing Association Annual Summit and Conference*, Hollywood, CA, 2012, pp. 1-6.

[7] Y.-H. Peng, H.-T. Hwang, Y.-C. Wu, Y. Tsao, and H.-M. Wang, "Exemplar-Based Spectral Detail Compensation for Voice Conversion," in *Proc.* INTERSPEECH, 2018, pp. 486–490.

[8] S. Ding, C. Liberatore, and R. Gutierrez-Osuna, "Learning Structured Dictionaries for Exemplar-based Voice Conversion," in *Proc. INTERSPEECH*, 2018, pp. 481–485.

[9] R. Li, Z. Wu, Y. Ning, L. Sun, H. Meng, and L. Cai, "Spectro-Temporal Modelling with Time-Frequency LSTM and Structured Output Layer for Voice Conversion," in *Proc. INTERSPEECH*, 2017, pp. 3409-3413.

[10] S. Liu, J. Zhong, L. Sun, X. Wu, X. Liu, and H. Meng, "Voice Conversion Across Arbitrary Speakers Based on a Single Target-Speaker Utterance," in Proc. *INTERSPEECH*, 2018, pp. 496–500.

[11] C. Hsu, H. Hwang, Y. Wu, Y. Tsao and H. Wang, "Voice conversion from non-parallel corpora using variational auto-encoder," *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, Jeju, 2016, pp. 1-6.

[12] C.-C. Hsu, H.-T. Hwang, Y.-C. Wu, Y. Tsao, and H.-M. Wang, "Voice conversion from unaligned corpora using variational autoencoding Wasserstein generative adversarial networks," in *Proc. INTERSPEECH*, 2017, pp. 3164–3168.

[13] J.-C. Chou, C.-C. Yeh, H.-Y. Lee, and L.-S. Lee, "Multi-target Voice Conversion without Parallel Data by Adversarially Learning Disentangled Audio Representations," in *Proc. INTERSPEECH*, 2018, pp. 501-505.

[14] T. Kaneko and H. Kameoka, "CycleGAN-VC: Non-parallel Voice Conversion Using Cycle-Consistent Adversarial Networks," *2018 26th European Signal Processing Conference*, Rome, 2018, pp. 2100-2104.

[15] K. Kobayashi, T. Toda, G. Neubig, S. Sakti, and S. Nakamura, "Statistical singing voice conversion with direct waveform modification based on the spectrum differential," in *Proc. INTERSPEECH*, 2014, pp. 2514-2518.

[16] K. Kobayashi, T. Toda, G. Neubig, S. Sakti, and S. Nakamura, "Statistical singing voice conversion based on direct waveform modification with global variance," in *Proc. INTERSPEECH*, 2015, pp. 2754-2758.

[17] Y. Choi, M. Choi, M. Kim, J. Ha, S. Kim and J. Choo, "StarGAN: Unified Generative Adversarial Networks for Multi-domain Image-to-Image Translation," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, 2018, pp. 8789-8797.

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

[19] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio. Generative adversarial nets. In NIPS, 2014.

[20] H. Zhang, I. J. Goodfellow, D. N. Metaxas, and A. Odena, "Self-Attention Generative Adversarial Networks," *arXiv preprint arXiv:1805.08318*, 2018.

[21] P. Luo, J. Ren, and Z. Peng, "Differentiable learning-to-normalize via switchable normalization," *arXiv preprint arXiv:1806.10779*, 2018.

[22] J. Pons, O. Slizovskaia, R. Gong, E. Gómez, and X. Serra, "Timbre analysis of music audio signals with convolutional neural networks," in *Proc. 25th Eur. Signal Process. Conf.*, pp. 2744-2748, 2017.

[23] J. Pons and X. Serra, "Designing efficient architectures for modeling temporal features with convolutional neural networks," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing*, New Orleans, LA, 2017, pp. 2472-2476.

[24] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[25] L. Mescheder, A. Geiger, and S. Nowozin. Which training methods for gans do actually converge? In International Conference on Machine Learning (ICML), 2018.

[26] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In ICML, 2015.
[27] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. arXiv:1607.08022, 2016.
[28] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. arXiv:1607.06450, 2016.
[29] T. Toda, L.-H. Chen, D. Saito, F. Villavicencio, M. Wester, Z. Wu, and J. Yamagishi, "The Voice Conversion Challenge 2016," in *Proc. INTERSPEECH*, 2016, pp. 1632–1636.
[30] A. Jansson, E. J. Humphrey, N. Montecchio, R. M. Bittner, A. Kumar, and T. Weyde, "Singing voice separation with deep u-net convolutional networks," in Proc. of ISMIR, 2017
[31] 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.
[32] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu. Deep speaker: an end-to-end neural speaker embedding system. CoRR, abs/1705.02304, 2017.
[33] C. Zhang, & K. Koishida. End-to-End Text-Independent Speaker Verification with Triplet Loss on Short Utterances. 10.21437/Interspeech.2017-1608.