SYNTHETIC-TO-NATURAL SPEECH WAVEFORM CONVERSION USING CYCLE-CONSISTENT ADVERSARIAL NETWORKS

Kou Tanaka, Takuhiro Kaneko, Nobukatsu Hojo, and Hirokazu Kameoka

NTT Communication Science Laboratories, NTT Corporation, Japan
Email: {tanaka.ko, kaneko.takuhiro, hojo.nobukatsu, kameoka.hirokazu}@lab.ntt.co.jp

ABSTRACT

We propose a learning-based filter that allows us to directly modify a synthetic speech waveform into a natural speech waveform. Speech-processing systems using a vocoder framework such as statistical parametric speech synthesis and voice conversion are convenient especially for a limited number of data because it is possible to represent and process interpretable acoustic features over a compact space, such as the fundamental frequency (F_0) and mel-cepstrum. However, a well-known problem that leads to the quality degradation of generated speech is an over-smoothing effect that eliminates some detailed structure of generated/converted acoustic features. To address this issue, we propose a synthetic-to-natural speech waveform conversion technique that uses cycle-consistent adversarial networks and which does not require any explicit assumption about speech waveform in adversarial learning. In contrast to current techniques, since our modification is performed at the waveform level, we expect that the proposed method will also make it possible to generate "vocoder-less" sounding speech even if the input speech is synthesized using a vocoder framework. The experimental results demonstrate that our proposed method can 1) alleviate the over-smoothing effect of the acoustic features despite the direct modification method used for the waveform and 2) greatly improve the naturalness of the generated speech sounds.

Index Terms— Statistical parametric speech synthesis, postfilter, deep neural network, generative adversarial network, cycle-consistent adversarial network

1. INTRODUCTION

Speech processing systems such as statistical parametric speech synthesis [1] and statistical voice conversion [2] are well-known frameworks. These approaches using a vocoder framework have a significant advantage, especially for a limited number of data, because it is possible to represent interpretable acoustic features over a compact space, such as the fundamental frequency (F_0) and mel-cepstrum, which are lower dimensional acoustic features than a short-term Fourier transform (STFT) spectrogram. Although these systems aim to produce speech with a quality indistinguishable from that of clean and real speech, processed and synthesized speech can usually be distinguished from natural speech. The realization of synthetic-to-natural speech waveform conversion provides significant benefit with many speech processing approaches, especially when using a vocoder framework. Three major factors reported in [3] degrade the speech synthesized by a statistical parametric speech synthesis technique: the accuracy of acoustic models, over-smoothing, which eliminates some detailed structure of generated/converted acoustic features, and vocoding. In this paper, we focus on vocoding and over-smoothing.

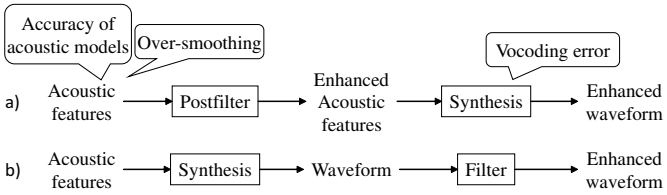

Fig. 1. Three major factors [3] that degrade the quality of synthesized speech during statistical parametric speech synthesis and general approaches to generating more natural sounding speech by using post-processing. Our proposed framework is assigned to a process b) which can address not only the over-smoothing problem but also the vocoding error.

To address the over-smoothing effect, several techniques for restoring the fine structure of natural speech over acoustic features have been proposed [2, 4, 5]. These approaches, as shown in Fig. 1 a), have achieved significant improvements as regards the naturalness of synthesized speech in the respective directions. However, heuristics approaches such as enhancement of global variance [2] and modulation spectrum [4] are unsuitable for covering all the negative factors. On the other hand, although a learning-based postfilter [5] enables us to restore not only the global variance and modulation spectrum but also other factors that degrade the quality of synthesized speech, it is still insufficient to generate natural speech because of the post filter needed not for the waveform but for the heuristic acoustic features such as mel-cepstrum. Furthermore, all of these approaches suffer from vocoding error be-

cause of the use of the vocoder framework to synthesize the speech waveform.

To avoid this limitation, an end-to-end speech enhancement [6] method has been proposed within a generative adversarial framework. As shown in Fig. 1 b), since the waveform of the input speech was directly operated to obtain that of the desired speech after the vocoding part, [6] has the potential to address not only the over-smoothing effect but also the vocoding error. Furthermore, the generative adversarial framework does not require us to design any hand-crafted feature that creates a gap between natural speech and synthetic speech, in advance. In preliminary experiments, we found that this method is unsuitable when the alignments [1] between the input waveform and the desired waveform are not perfect. For example, the noise reduction of noisy speech simulated by adding noise to the speech waveform recorded in an ideal environment succeeded because of the perfect alignment between the simulated noisy speech and the clean source speech. However, the conversion of synthetic speech generated by text-to-speech synthesis and voice conversion processing to natural speech is not easy to achieve by applying this method because of the alignment problem as mentioned above.

In this paper, we propose a learning-based filter that allows us to convert a synthetic speech waveform into a natural speech waveform using cycle-consistent adversarial networks with a fully convolutional architecture. We adopt cycle-consistent adversarial networks because they do not require a dataset forcibly paired at the time frame level and as the name implies, they are trained within the adversarial learning. In contrast to [7] which is also inspired by the cycle-consistent adversarial networks [8] to convert not speech waveform but acoustic feature, since our modification is performed at the waveform level, we expect that the proposed method will make it possible to generate "vocoderless" sounding speech even if the input speech is synthesized using a vocoder framework. Furthermore, we adopt a gated convolutional neural network (CNN) architecture [9], which is able to capture long- and short-term dependencies in the speech waveform. The experimental results demonstrate that our proposed method can 1) alleviate the over-smoothing effect of the acoustic features despite the direct modification method used for the waveform and 2) greatly improve the naturalness of the generated speech sounds.

2. SEGAN: SPEECH ENHANCEMENT GENERATIVE ADVERSARIAL NETWORK

2.1. Generative Adversarial Networks

Generative Adversarial Networks (GANs) [10] are generative models consisting of two neural networks. One is a generator G that learns to convert a sample $\boldsymbol{z}$ from a prior distribution $P(\boldsymbol{z})$ to a target sample $\boldsymbol{x}$ from a distribution $P_{\mathrm{Data}(\boldsymbol{x})}$, which is a sample from the training data. The generator aims to learn a projection that can imitate the true feature distribution and to generate samples related to the training data. The other is a discriminator D that learns the boundary between imitated features generated by the generator G and true features picked up from the training data.

The adversarial characteristic arises from the fact that the discriminator D tries to classify the instances $\boldsymbol{x}$ obtained from the true data distribution $P_{\mathrm{Data}}(\boldsymbol{x})$ as real and the candidates $G(\boldsymbol{z})$ produced by the generator G as fake, while the generator G tries to make the discriminator D classify those $G(\boldsymbol{z})$ as real. Through back-propagation, the generator G becomes able to generate better candidates $G(\boldsymbol{z})$ and the discriminator D becomes able to distinguish the generated ones $G(\boldsymbol{z})$ and real data $\boldsymbol{x}$. The objective function of the adversarial learning is formulated as the following minimax game between G and D,

$$\begin{aligned}\min_G \max_D \mathcal{L}_{\mathrm{gan}}(G_{\boldsymbol{z}\to\boldsymbol{x}}, D_{\boldsymbol{x}}) \\ = \mathbb{E}_{x\sim P_{\mathrm{Data}(x)}}[\log D_{\boldsymbol{x}}(\boldsymbol{x})] \\ + \mathbb{E}_{z\sim P_z(z)}[\log(1 - D_{\boldsymbol{x}}(G_{\boldsymbol{z}\to\boldsymbol{x}}(\boldsymbol{z})))]. \quad (1)\end{aligned}$$

Although the GANs achieve state-of-the-art results in a variety of generative tasks [11, 12], the difficulty of the training is a well-known problem. For instance, the classic approach suffers from a vanishing gradient problem due to the sigmoid cross-entropy loss used for training. Several adversarial training techniques have been proposed to overcome this difficulty. The least-squares GAN (LSGAN) approach [13] stabilizes the training process by replacing the cross-entropy loss shown in Eq. 1 with the least-squares function as follows.

$$\begin{aligned}\min_D \mathcal{L}_{\mathrm{lsgan}}(D_{\boldsymbol{x}}) = \frac{1}{2}\mathbb{E}_{x\sim P_{\mathrm{Data}(x)}}[(D_{\boldsymbol{x}}(\boldsymbol{x}) - 1)^2] \\ + \frac{1}{2}\mathbb{E}_{z\sim P_z(z)}[D_{\boldsymbol{x}}(G_{\boldsymbol{z}\to\boldsymbol{x}}(\boldsymbol{z}))^2], \quad (2)\end{aligned}$$

$$\min_G \mathcal{L}_{\mathrm{lsgan}}(G_{\boldsymbol{z}\to\boldsymbol{x}}) = \frac{1}{2}\mathbb{E}_{z\sim P_z(z)}[(D_{\boldsymbol{x}}(G_{\boldsymbol{z}\to\boldsymbol{x}}(\boldsymbol{z})) - 1)^2]. \quad (3)$$

2.2. GANs for Speech Enhancement

To retain the linguistic information of speech samples, [6] adopts a conditioned version GAN that has some extra information in G and D to perform mapping and classification. As shown in Fig. 2, in the structure of the generator G, which is similar to an auto-encoder, a noisy speech signal $\boldsymbol{x}_{\mathrm{n}}$, which is the input of the G network, is encoded as $\boldsymbol{x}_c$. After concatenating the random vector $\boldsymbol{z}$ with the encoded vector $\boldsymbol{x}_c$, which is treated as a conditional vector, the decoding part of the G network is performed as transposed convolutions (a.k.a.

[1] In this paper, we define alignment considering both the magnitude information and the phase information of speech because we focus on modifying the speech waveform rather than the acoustic features.

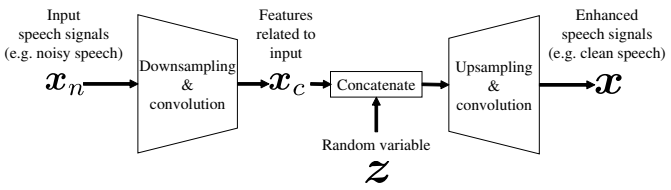

Fig. 2. Generator network for speech enhancement reported in [6]. Structure is similar to an auto-encoder.

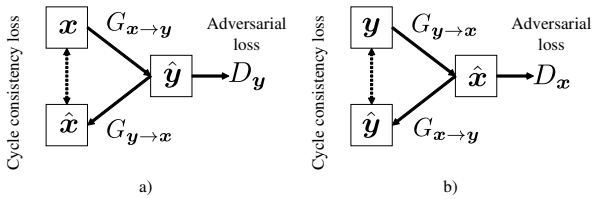

Fig. 3. Training procedures of cycle-consistent adversarial networks: a) Forward-inverse mapping to consider forward cycle consistency and b) inverse-forward mapping to consider backward cycle consistency.

deconvolutions or fractionally strided convolutions) to obtain the enhanced vector.

To achieve the generation of speech samples that are closer to clean speech, a secondary component is added to the loss of G. [6] adopts the L1 norm, as it has been proven to be effective in the image manipulation domain [14, 15]. In this way, they allow the adversarial component to add more fine-grained and realistic results. A new hyper-parameter λ_{SEGAN} controls the magnitude of the L1 norm. Finally, the loss function of the generator G becomes

$$\begin{aligned}&\min_G \mathcal{L}_{\mathrm{lsgan}}(G_{\boldsymbol{x}_\mathrm{n},\boldsymbol{z}\to\boldsymbol{x}})\\&= \frac{1}{2}\mathbb{E}_{z\sim P_z(z),x_\mathrm{n}\sim P_{Data(x_\mathrm{n})}}[(D_{\boldsymbol{x}}(G_{\boldsymbol{x}_\mathrm{n},\boldsymbol{z}\to\boldsymbol{x}}(\boldsymbol{x}_\mathrm{n},\boldsymbol{z}))-1)^2]\\&+\lambda_{\mathrm{SEGAN}}||G_{\boldsymbol{x}_\mathrm{n},\boldsymbol{z}\to\boldsymbol{x}}(\boldsymbol{x}_\mathrm{n},\boldsymbol{z})-\boldsymbol{x}||_1. \qquad (4)\end{aligned}$$

3. SYNTHETIC-TO-NATURAL SPEECH WAVEFORM CONVERSION USING CYCLE-CONSISTENT ADVERSARIAL NETWORKS

3.1. Concept

In preliminary experiments, we found that SEGAN [6] could not be easily applied to the conversion of a synthetic speech waveform to a natural speech waveform. One possible reason is that the misalignment caused by the different lengths and generation processes of synthetic and natural speech makes it difficult to ensure the operation of the bijective function in the generator G. Specifically, the phase information of the speech waveform synthesized by using the vocoder framework is very far from that of natural speech, even if the magnitude information of the synthetic speech is close to that of natural speech. We assume that these factors induce "mode collapse", which is a well-known problem when training GANs, and the SEGAN does not guarantee that an individual input and output are paired up in a meaningful way. Generally speaking, all input speech signals map to the same output speech signals and the optimization fails to make progress [10].

To solve this problem, we focus on cycle-consistent adversarial networks [8]. This approach has introduced a "cycle consistent" property, which ensures return to the original sample [16]. Mathematically, if we have a converter $G_{\boldsymbol{x}\to\boldsymbol{y}}$ and another converter $G_{\boldsymbol{y}\to\boldsymbol{x}}$, $G_{\boldsymbol{x}\to\boldsymbol{y}}$ and $G_{\boldsymbol{y}\to\boldsymbol{x}}$ should be the inverse of each other, and both mappings should be bijections. We incorporate this property into SEGAN by training the mapping functions $G_{\boldsymbol{x}\to\boldsymbol{y}}$ and $G_{\boldsymbol{y}\to\boldsymbol{x}}$ simultaneously and adding a cycle consistency loss [17] that encourages $G_{\boldsymbol{y}\to\boldsymbol{x}}(G_{\boldsymbol{x}\to\boldsymbol{y}}(\boldsymbol{x}))\approx\boldsymbol{x}$ and $G_{\boldsymbol{x}\to\boldsymbol{y}}(G_{\boldsymbol{y}\to\boldsymbol{x}}(\boldsymbol{y}))\approx\boldsymbol{y}$. Combining the cycle consistency loss with the adversarial losses defines our full objective for a training procedure using perfect alignment.

Furthermore, we focus on a convolutional architecture called a gated CNN. The gated CNN has recently been shown to be powerful for modeling long-term sequential data. It was originally introduced for language modeling and was shown to outperform long short-term memory (LSTM) language models trained in a similar setting [9]. We previously applied a gated CNN architecture for acoustic feature sequence modeling, and its effectiveness has already been confirmed [18, 19]. With a gated CNN, the output of a hidden layer of a network is described as a linear projection modulated by an output gate. Similar to an LSTM [20] and a gated recurrent unit (GRU) [21], the output gate controls what information should be propagated through the hierarchy of layers and allows the capture of long-term structures.

3.2. Cycle-Consistent Adversarial Networks

For each speech sample $\boldsymbol{x}$, the speech waveform conversion cycle shown in Fig. 3 a) constrains the samples $\boldsymbol{x}$ to return to the original speech through a target domain corresponding to the samples $\boldsymbol{y}$, $\boldsymbol{x}\to G_{\boldsymbol{x}\to\boldsymbol{y}}(\boldsymbol{x})\to G_{\boldsymbol{y}\to\boldsymbol{x}}(G_{\boldsymbol{x}\to\boldsymbol{y}}(\boldsymbol{x}))\approx\boldsymbol{x}$. This cycle consistency is called forward cycle consistency. Similarly, as shown in Fig. 3 b), for each speech waveform $\boldsymbol{y}$, $G_{\boldsymbol{x}\to\boldsymbol{y}}$ and $G_{\boldsymbol{y}\to\boldsymbol{x}}$ are constrained by a backward cycle consistency, $\boldsymbol{y}\to G_{\boldsymbol{y}\to\boldsymbol{x}}(\boldsymbol{y})\to G_{\boldsymbol{x}\to\boldsymbol{y}}(G_{\boldsymbol{y}\to\boldsymbol{x}}(\boldsymbol{y}))\approx\boldsymbol{y}$. Therefore, these are described as the following cycle consistency loss,

$$\begin{aligned}\mathcal{L}_{\mathrm{cyc}} &= \mathbb{E}_{x\sim P_{\mathrm{Data}(x)}}[||G_{\boldsymbol{y}\to\boldsymbol{x}}(G_{\boldsymbol{x}\to\boldsymbol{y}}(\boldsymbol{x}))-\boldsymbol{x}||_1]\\&+\mathbb{E}_{y\sim P_{\mathrm{Data}(y)}}[||G_{\boldsymbol{x}\to\boldsymbol{y}}(G_{\boldsymbol{y}\to\boldsymbol{x}}(\boldsymbol{y}))-\boldsymbol{y}||_1]. \qquad (5)\end{aligned}$$

Finally, the objective function is

$$\mathcal{L}_{\text{full}} = \mathcal{L}_{\text{gan}}(G_{\boldsymbol{x}\to\boldsymbol{y}}, D_{\boldsymbol{y}}) + \mathcal{L}_{\text{gan}}(G_{\boldsymbol{y}\to\boldsymbol{x}}, D_{\boldsymbol{x}}) + \lambda_{\text{cyc}}\mathcal{L}_{\text{cyc}}, \quad (6)$$

where λ_{cyc} is a hyper parameter used to control the cycle consistency loss.

3.3. Identity-Mapping Loss

Cycle consistency loss allows us to reduce the possible mapping functions by constraining a structure. However, in a waveform modification task, the linguistic information is not always preserved by incorporating only the cycle consistency loss. The identity-mapping loss reported in [22] preserves the compositions of the input samples and the converted samples. [8] has applied this approach to color preservation and demonstrated its effectiveness. Note that the secondary component of Eq. 4 is also identity-mapping loss. To encourage the generators $G_{\boldsymbol{x}\to\boldsymbol{y}}$ and $G_{\boldsymbol{y}\to\boldsymbol{x}}$ to preserve linguistic information, we also incorporate this property as follows.

$$\mathcal{L}_{\text{id}} = \mathbb{E}_{x\sim P_{\text{Data}(x)}}[||G_{\boldsymbol{y}\to\boldsymbol{x}}(\boldsymbol{x}) - \boldsymbol{x}||_1] + \mathbb{E}_{y\sim P_{\text{Data}(y)}}[||G_{\boldsymbol{x}\to\boldsymbol{y}}(\boldsymbol{y}) - \boldsymbol{y}||_1]. \quad (7)$$

In practice, the weighted loss $\lambda_{\text{id}}\mathcal{L}_{\text{id}}$ with a hyper parameter λ_{id} to control the identity-mapping loss is added to Eq. 6.

3.4. Sequential Modeling with Gated CNN

To capture long- and short-term dependencies in speech waveforms, we use a gated CNN [9] to construct both the generator and discriminator networks of the GAN. The gated CNNs are CNNs equipped with gated linear units (GLUs) as activation functions instead of the regular rectified linear units (ReLUs) [23] or Tanh activations. The output of the l_{th} hidden layer of a gated CNN is described as a linear projection $\boldsymbol{H}_{l-1} * \boldsymbol{W}_l + \boldsymbol{b}_l$ modulated by an output gate $\sigma(\boldsymbol{H}_{l-1} * \boldsymbol{V}_l + \boldsymbol{c}_l)$

$$\boldsymbol{H}_l = (\boldsymbol{H}_{l-1} * \boldsymbol{W}_l + \boldsymbol{b}_l) \otimes \sigma(\boldsymbol{H}_{l-1} * \boldsymbol{V}_l + \boldsymbol{c}_l), \quad (8)$$

where $\boldsymbol{W}_l$, $\boldsymbol{V}_l$, $\boldsymbol{b}_l$ and $\boldsymbol{c}_l$ are the network parameters to be trained, σ is the sigmoid function and $\otimes$ indicates the element-wise product. Similar to LSTMs, the output gate multiplies each element of $\boldsymbol{H}_{l-1} * \boldsymbol{W}_l + \boldsymbol{b}_l$ and controls what information should be propagated through the hierarchy of layers in a data-driven manner.

4. EXPERIMENTAL EVALUATION

4.1. Experimental Conditions

Datasets (Natural): We used a Japanese speech dataset consisting of utterances by one professional female narrator. To evaluate the performance, we used 30 sentences (speech sections of 5.3 minutes). To train the models, we used about 6,500 sentences for a baseline system and 400 sentences (speech sections of 1.2 hours) for the conventional and proposed methods. The sampling rate of the speech signals was 22.05 kHz. Audio samples can be accessed on our web page[2].

Baseline system (Baseline): We used a DNN-based statistical parametric speech synthesis method [1] as the baseline. From the speech data, 40 Mel-cepstral coefficients, logarithmic F_0, and 5-band aperiodicities were extracted every 5 ms with the STRAIGHT analysis system [24, 25]. The contextual features used as the input were 506-dimensional linguistic features including phonemes and mora positions. The output consisted of 40 Mel-cepstral coefficients, log F_0, 5-band aperiodicities, their delta and delta-delta features, and a voiced/unvoiced binary value. The DNN architectures were feed-forward networks including 5 hidden layers each with 1,024 units.

Conventional method (GANv): As a conventional approach, we used a GAN-based postfilter [5] not for the speech waveform but for the acoustic features. The system setting was the same as the reported setting, except for the excitation signals. Although [5] used the excitation signals of natural speech, we used the excitation signals generated by the vocoding for evaluating all of the synthetic speech. We applied the conventional method only to voiced segments.

Our proposed method (Proposed): We designed a network based on recent success of image modeling [26]. Figure 4 shows the network architectures of our proposed model. The network included downsampling layers, residual blocks [27], and upsampling layers. We used instance normalization (IN) [28], instead of batch normalization [29]. We used pixel shuffler (PS) for upsampling where the effectiveness was demonstrated in high-resolution image generation [26]. We normalized the speech waveform to zero mean and unit variance using their training sets. To stabilize the training, we used a least squares GAN [13]. We set λ_{cyc} at 10. To guide the learning process, we set λ_{id} at 5 for the first 20k iterations and linearly decay to 0 over the next 20k iterations. We optimized the model parameters using the Adam optimizer [30] with a mini-batch of size 32. The learning parameters α were set at 0.0001 for discriminators and 0.0002 for generators. We used the same learning rate for the first 250k iterations and linearly decay to 0 over the next 250k iterations. The other learning parameters of the Adam optimizer, β_1 and β_2, were set at 0.5 and 0.99, respectively. Note that since the generators are fully convolutional, they can handle an arbitrary length input.

[2] http://www.kecl.ntt.co.jp/people/tanaka.ko/projects/s2n/s2n_speech_waveform_conversion.html

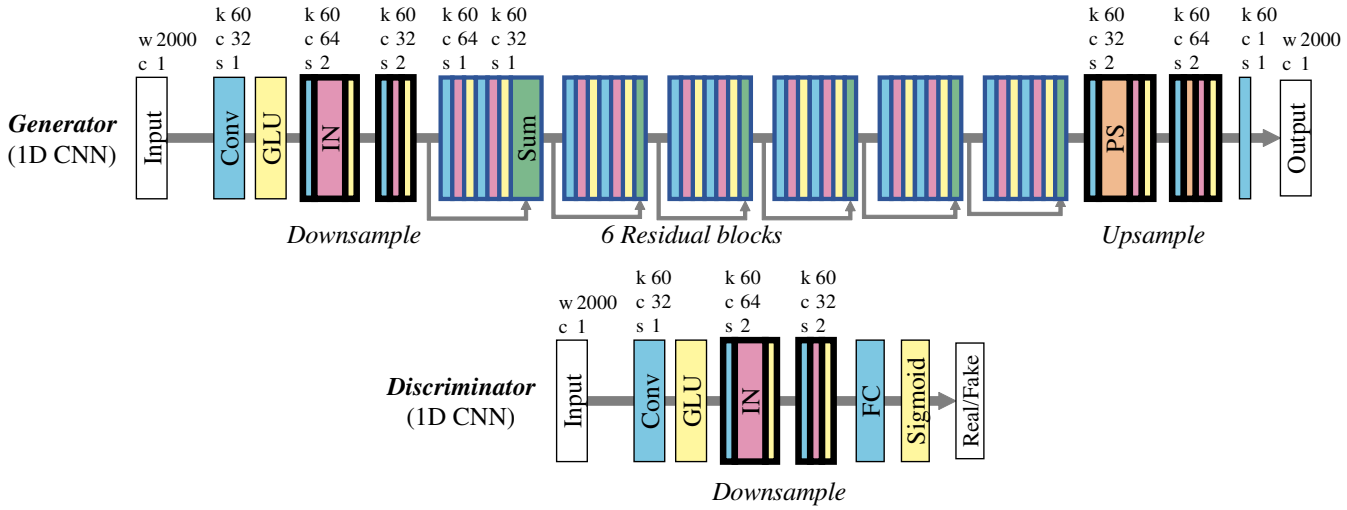

Fig. 4. Network architectures of generator G and discriminator D. "Conv", "GLU", "IN", "PS", "FC", and "Sigmoid" denote convolution, instance normalization, gated linear unit, pixel shuffler, fully connected, and sigmoid layers, respectively. In an input or output layer, w and c represent width and number of channels, respectively. In each convolutional layer, k, c, and s denote kernel size, number of channels, and stride size, respectively.

4.2. Modulation Spectrum over Acoustic Features

To confirm the alleviation of the over-smoothing effect of the acoustic features, we applied the conventional and proposed methods to speech synthesized by the baseline system and obtained modulation spectrums of mel-cepstrum sequences on each system. Although the modulation spectrum is traditionally defined as a value calculated using the Fourier transform of the parameter sequence [31], this paper defines the modulation spectrum as its logarithmic power spectrum. We used 8,192 FFT points.

The average modulation spectrums of the first 1k indices for the 10th, 20th, 30th and 40th mel-cepstral coefficient sequences are shown in Fig. 5. We found that **Baseline** suffered more from the over-smoothing effect than **GANv** and **Natural**. On the other hand, **GANv** and **Proposed** are close to **Natural**. As with the GAN-based postfilter for the acoustic feature **GANv**, the result demonstrated that our proposed method for the speech waveform **Proposed** successfully alleviated the over-smoothing effect caused by the statistical parametric speech synthesis process.

4.3. Subjective Evaluation for Naturalness

We conducted a subjective 5-scale mean opinion score test regarding the naturalness of the generated speech. 10 listeners participated and each listener evaluated 120 speech samples (30 speech samples $\times$ 4 systems). We applied the conventional and proposed methods to the same speech waveform **Baseline** as in Sec. 4.2.

Figure 6 shows that our proposed method **Proposed** achieved a significant improvement in terms of the naturalness of the generated speech, compared with **Baseline** and **GANv**. This result indicates that our approach is more effective than the use of postfilters for the acoustic features because it is possible to address both the over-smoothing problem and the vocoding error. Furthermore, with **Proposed**, the listeners commented that the "buzzy" sound peculiar to vocoding was sufficiently improved. However, there is still a gap between **Proposed** and **Natural**. One possible reason for the gap is the " hoarse " sound of **Proposed**. The listeners also advised that **Proposed** was distinguishable from **Natural** because **Proposed** sometimes had a "hoarse" sound.

5. CONCLUSION

In this paper, to realize a synthetic-to-natural speech filter, we proposed a learning-based filter that allows us to convert a synthetic speech waveform into a natural speech waveform using cycle-consistent adversarial networks. Since our process was applied after the synthesis part in statistical parametric speech synthesis, we expected that our approach would be able to address not only the over-smoothing problem but also the vocoding error. The experimental results demonstrated that our proposed method 1) alleviated the over-smoothing effect of the acoustic features despite the direct modification method used for the waveform and 2) dramatically improved the naturalness of the generated speech sounds. In the future, we will further fill the gap between natural speech and synthetic speech by considering the auditory property.

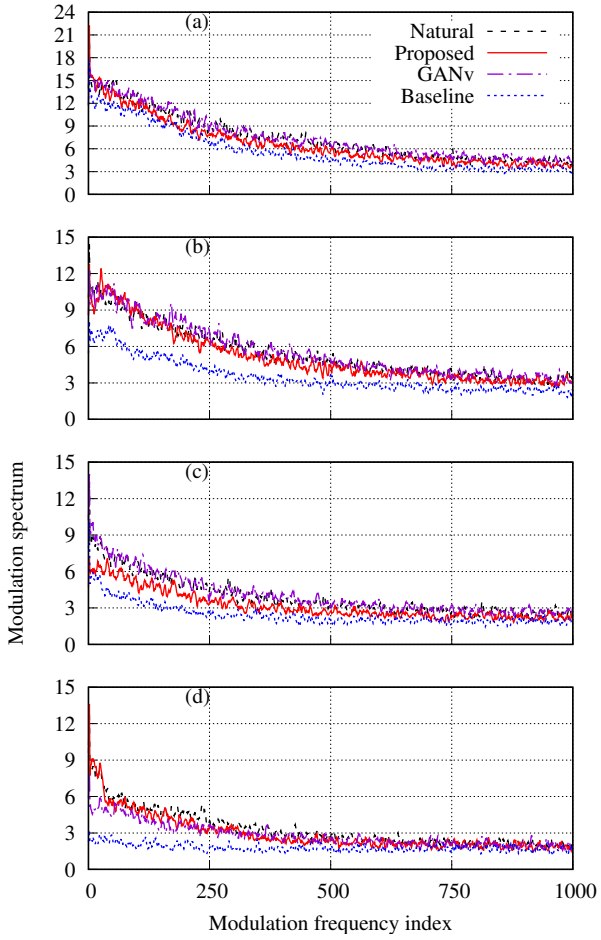

Fig. 5. Average modulation spectrums of the first 1k indices for a) 10th, b) 20th, c) 30th and d) 40th mel-cepstral coefficient sequences.

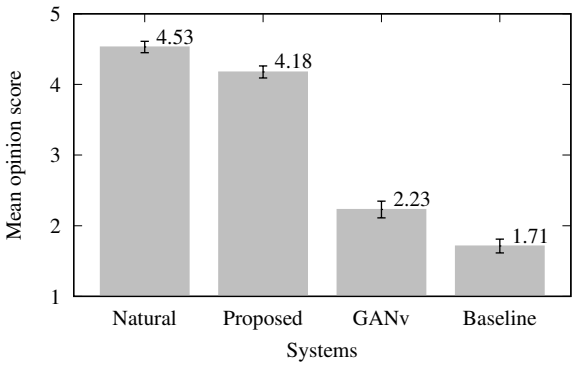

Fig. 6. Subjective 5-scale mean opinion score regarding naturalness, with 95% confidence intervals.

Acknowledgment

This work was supported by JSPS KAKENHI 17H01763.

6. REFERENCES

[1] Heiga Zen, Andrew Senior, and Mike Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7962–7966. 1, 4

[2] Tomoki Toda, Alan W Black, and Keiichi 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. 1

[3] Heiga Zen, Keiichi Tokuda, and Alan W Black, "Statistical parametric speech synthesis," *Speech Communication*, vol. 51, no. 11, pp. 1039–1064, 2009. 1

[4] Shinnosuke Takamichi, Tomoki Toda, Graham Neubig, Sakriani Sakti, and Satoshi 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. 1

[5] Takuhiro Kaneko, Hirokazu Kameoka, Nobukatsu Hojo, Yusuke Ijima, Kaoru Hiramatsu, and Kunio Kashino, "Generative adversarial network-based postfilter for statistical parametric speech synthesis," in *Proc. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2017)*, 2017, pp. 4910–4914. 1, 4

[6] Santiago Pascual, Antonio Bonafonte, and Joan Serrà, "SEGAN: Speech enhancement generative adversarial network," *arXiv preprint arXiv:1703.09452*, 2017. 2, 3

[7] Takuhiro Kaneko and Hirokazu Kameoka, "Parallel-data-free voice conversion using cycle-consistent adversarial networks," *arXiv preprint arXiv:1711.11293*, 2017. 2

[8] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," *arXiv preprint arXiv:1703.10593*, 2017. 2, 3, 4

[9] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier, "Language modeling with gated convolutional networks," *arXiv preprint arXiv:1612.08083*, 2016. 2, 3, 4

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron

Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680. 2, 3

[11] Alec Radford, Luke Metz, and Soumith Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015. 2

[12] Yuki Saito, Shinnosuke Takamichi, Hiroshi Saruwatari, Yuki Saito, Shinnosuke Takamichi, and Hiroshi Saruwatari, "Statistical parametric speech synthesis incorporating generative adversarial networks," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 26, no. 1, pp. 84–96, 2018. 2

[13] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley, "Least squares generative adversarial networks," in *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017, pp. 2813–2821. 2, 4

[14] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros, "Image-to-image translation with conditional adversarial networks," *arXiv preprint*, 2017. 3

[15] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros, "Context encoders: Feature learning by inpainting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2536–2544. 3

[16] Richard W Brislin, "Back-translation for cross-cultural research," *Journal of cross-cultural psychology*, vol. 1, no. 3, pp. 185–216, 1970. 3

[17] Tinghui Zhou, Philipp Krahenbuhl, Mathieu Aubry, Qixing Huang, and Alexei A Efros, "Learning dense correspondence via 3D-guided cycle consistency," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 117–126. 3

[18] Takuhiro Kaneko, Hirokazu Kameoka, Kaoru Hiramatsu, and Kunio Kashino, "Sequence-to-sequence voice conversion with similarity metric learned using generative adversarial networks," *Proc. Interspeech 2017*, pp. 1283–1287, 2017. 3

[19] Kou Tanaka, Hirokazu Kameoka, and Kazuho Morikawa, "VAE-SPACE: Deep generative model of voice fundamental frequency contours," *Proc. ICASSP 2018*, 2018. 3

[20] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997. 3

[21] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014. 3

[22] Yaniv Taigman, Adam Polyak, and Lior Wolf, "Unsupervised cross-domain image generation," *arXiv preprint arXiv:1611.02200*, 2016. 4

[23] Vinod Nair and Geoffrey E Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814. 4

[24] Hideki Kawahara, Ikuyo Masuda-Katsuse, and Alain De Cheveigne, "Restructuring speech representations using a pitch-adaptive time–frequency smoothing and an instantaneous-frequency-based f0 extraction: Possible role of a repetitive structure in sounds," *Speech communication*, vol. 27, no. 3, pp. 187–207, 1999. 4

[25] Hideki Kawahara, Jo Estill, and Osamu Fujimura, "Aperiodicity extraction and control using mixed mode excitation and group delay manipulation for a high quality speech analysis, modification and synthesis system straight," in *Second International Workshop on Models and Analysis of Vocal Emissions for Biomedical Applications*, 2001. 4

[26] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al., "Photo-realistic single image super-resolution using a generative adversarial network," *arXiv preprint*, 2016. 4

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778. 4

[28] Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *CoRR*, vol. abs/1607.08022, 2016. 4

[29] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, 2015, pp. 448–456. 4

[30] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. 4

[31] Les Atlas and Shihab A. Shamma, "Joint acoustic and modulation frequency," *EURASIP Journal on Advances in Signal Processing*, vol. 2003, no. 7, pp. 310290, June 2003. 5