SINGING VOICE SYNTHESIS BASED ON GENERATIVE ADVERSARIAL NETWORKS

Yukiya Hono, Kei Hashimoto, Keiichiro Oura, Yoshihiko Nankaku, and Keiichi Tokuda

Department of Computer Science, Nagoya Institute of Technology, Nagoya, Japan

ABSTRACT

This paper proposes a generative adversarial training method for deep neural network (DNN)-based singing voice synthesis. The DNN-based approach has been used in statistical parametric singing voice synthesis and improved the naturalness of the synthesized singing voice [1]. Recently, generative adversarial networks (GANs) [2] have attracted significant attention in various machine learning research areas including speech synthesis [3]. GANs have achieved great success in modeling the distributions of complex data, and they have the potential to alleviate over-smoothing problem on the generated speech parameters in speech synthesis. In this paper, we propose a DNN-based singing voice synthesis system incorporating the GAN. Experimental results show that the proposed method outperforms the conventional method in the naturalness of the synthesized singing voice.

Index Terms— Singing voice synthesis, generative adversarial network, neural network

1. INTRODUCTION

In recent years, deep neural networks (DNNs) have attained significant improvement in various machine learning areas such as image recognition [4, 5], speech recognition [6], and speech synthesis [7, 8]. In a DNN-based text-to-speech synthesis system, DNN-based acoustic models can represent complex dependencies between linguistic feature sequences and acoustic feature sequences more efficiently than conventional hidden Markov model (HMM)-based acoustic models [9]. DNN-based singing voice synthesis has also been proposed, and it can produce a natural sounding synthesized singing voice [1]. Recently, neural networks that can model audio waveforms directly, e.g., WaveNet [10], SampleRNN [11], and FFTNet [12], have been proposed. Such neural networks are used as vocoders in the speech field and improve the quality of synthesized speech compared to conventional vocoders. The neural vocoders use acoustic features as inputs. Therefore, accurately predicting acoustic features from linguistic features by acoustic models is still an important issue to generate high quality speech or singing voice.

DNN-based acoustic models are generally trained with the minimum mean squared error (MSE) criterion or maximum likelihood criterion. However, this is problematic for the prediction of acoustic features. It is known that the distribution of acoustic features is multimodal, as humans can sing the same lyrics in many different ways. The conventional training approaches of neural networks cannot learn to model any more complex distributions of acoustic features than a unimodal Gaussian distribution. Hence, the generated speech parameters tend to be over-smoothed, which leads to deterioration of the naturalness of synthesized speech.

Generative adversarial networks (GANs) [2] were recently introduced as a novel way to train a generative model. A GAN is a powerful generative model that has been successfully used in image generation [13, 14] and other tasks [15, 16]. They consist of two neural networks: a generator that captures the data distribution, and a discriminator that estimates the probability that a sample came from the training data rather than the generator. These networks are trained adversarially, where the generator aims at deceiving the discriminator, and the discriminator is trained to distinguish the natural and generate feature samples. GANs have achieved great success in modeling the distributions of complex data because GAN-based training is equivalent to minimizing the divergence between true data distribution and generated data distribution. In text-to-speech synthesis, a GAN-based training method has been proposed [3, 17]. The generator acts as an acoustic model and is optimized by an adversarial loss computed using the discriminator. It is reported that the over-smoothing effect of the generated speech parameters is alleviated, and the naturalness of synthesized speech is significantly improved by adversarial training.

In this paper, we introduce the generative adversarial network into the DNN-based singing voice synthesis system. Additionally, we propose a the DNN-based singing voice synthesis with the conditional generative adversarial network (CGAN) [18]. By applying GAN and CGAN-based training, the acoustic models are optimized for complex distributions representing the acoustic features of singing voices.

The rest of this paper is organized as follows. Sections 2 and 3 of this paper describe statistical parametric singing voice synthesis based on DNNs and the generative adversarial training method, respectively. The experimental conditions and results are provided in Section 4. Concluding remarks and future work are presented in Section 5.

2. CONVENTIONAL DNN-BASED SINGING VOICE SYNTHESIS

Figure 1 gives an overview of the DNN-based singing voice synthesis system [1]. In the statistical parametric singing voice synthesis using DNN-based acoustic models [1], a DNN represents a mapping function from score feature sequences including linguistic and musical score information (e.g., phonetic, note key, and note length features) to acoustic feature sequences (e.g., spectral, excitation, and vibrato parameters). In the training phase, the DNN aims to minimize the loss function $L(\boldsymbol{o}, \hat{\boldsymbol{o}})$ as

$$L(\boldsymbol{o}, \hat{\boldsymbol{o}}) = -\prod_{t=1}^{T} \mathcal{N}(\boldsymbol{o}_t \mid \hat{\boldsymbol{o}}_t, \boldsymbol{\Sigma}_t), \tag{1}$$

$$\hat{\boldsymbol{o}}_t = g(\boldsymbol{l}_t), \tag{2}$$

where T is the number of frames included in a song, $\boldsymbol{o}$ is a sequence of acoustic feature vectors consisting of a static and their dynamic feature vectors, $\hat{\boldsymbol{o}}$ is the output parameter from a trained neural network, $\boldsymbol{l}$ is a sequence of score feature vectors, and $g(\cdot)$ is a non-linear mapping function represented by the DNN. $\mathcal{N}(\cdot \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the

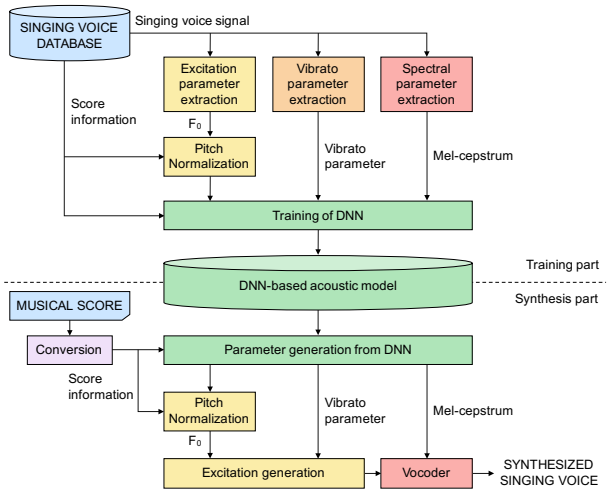

Fig. 1. Overview of the conventional DNN-based singing voice synthesis system.

Gaussian distribution with a mean vector $\boldsymbol{\mu}$ and a covariance matrix $\boldsymbol{\Sigma}$. In the synthesis phase, the score features extracted from a given musical score to be synthesized are mapped to acoustic features by the trained DNN. The optimal static-feature sequence $\hat{\boldsymbol{c}}$ is given by

$$\hat{\boldsymbol{c}} = \arg\max_{\boldsymbol{c}} \mathcal{N}(\boldsymbol{W}\boldsymbol{c} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{3}$$

where $\boldsymbol{c}$ is the static feature sequences and $\boldsymbol{W}$ is a window matrix for calculating dynamic features from a static feature sequence. Parameter trajectories are generated considering the relation between static and dynamic features by the maximum likelihood parameter generation (MLPG) algorithm [19] to generate smooth speech parameter trajectories.

In the statistical parametric approaches to singing voice synthesis, it is difficult to express contextual factors that hardly ever appear in the training data. Although databases including various contextual factors should be used in DNN-based singing voice synthesis systems, it is almost impossible to cover all possible contextual factors because singing voices involve a huge number of them, e.g., keys, lyrics, dynamics, note positions, durations, and pitch. Pitch should be correctly covered because generated F_0 trajectories greatly affect the quality of the synthesized singing voices.

To address this problem, a musical-note-level pitch normalization technique has been proposed for DNN-based singing voice synthesis systems [1]. In this technique, the differences between the log F_0 sequences extracted from waveforms and the pitch of musical notes are modeled. This technique makes it possible for DNN-based singing voice synthesis systems to generate variable singing voices that include any pitch.

3. ADVERSARIAL TRAINING FOR SINGING VOICE SYNTHESIS

A GAN is a framework for learning deep generative models by an adversarial process. It consists of two neural networks: a generator that captures the data distribution, and a discriminator that estimates the probability that a sample came from the training data rather than the generator. In this paper, the GAN is applied to the singing voice synthesis.

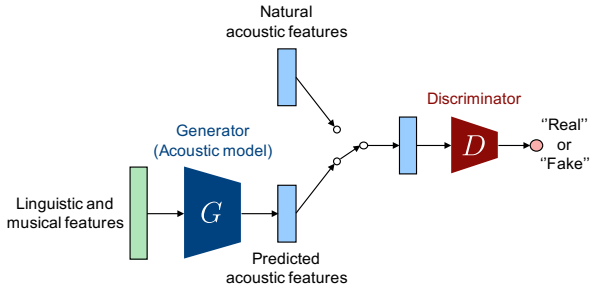

Fig. 2. The proposed GAN-based training framework for singing voice synthesis.

3.1. GAN-based singing voice synthesis

Figure 2 gives an overview of the GAN-based training framework for singing voice synthesis. In this framework, the generator G is an acoustic model that predicts acoustic features $\boldsymbol{o}_t$ from score feature sequences $\boldsymbol{l}_t$, and it is written by $\hat{\boldsymbol{o}}_t = G(\boldsymbol{l}_t)$. It should be noted that there is a difference from vanilla GAN [2] in that the inputs of the generator are score feature sequences instead of random noise. The discriminator is a classifier that aims to discriminate whether acoustic features are real features from the training data or fake features generated by G. $D(\boldsymbol{o}_t)$ represents the posterior probability that $\boldsymbol{o}_t$ came from the data rather than training data. The discriminator is optimized by minimizing the following loss function:

$$L_D^{\mathrm{GAN}}(\boldsymbol{o}, \hat{\boldsymbol{o}}) = L_{D,1}^{\mathrm{GAN}}(\boldsymbol{o}) + L_{D,0}^{\mathrm{GAN}}(\hat{\boldsymbol{o}}), \tag{4}$$

where $L_{D,1}^{\mathrm{GAN}}(\boldsymbol{o})$ and $L_{D,0}^{\mathrm{GAN}}(\hat{\boldsymbol{o}})$ represent loss functions for natural acoustic features and generated acoustic features, respectively, which are given by

$$L_{D,1}^{\mathrm{GAN}}(\boldsymbol{o}) = -\frac{1}{T}\sum_{t=1}^{T} \log D(\boldsymbol{o}_t), \tag{5}$$

$$L_{D,0}^{\mathrm{GAN}}(\hat{\boldsymbol{o}}) = -\frac{1}{T}\sum_{t=1}^{T} \log\bigl(1 - D(\hat{\boldsymbol{o}}_t)\bigr). \tag{6}$$

After updating the discriminator, the generator is trained to minimize the following loss function $L_G^{\mathrm{GAN}}(\boldsymbol{o}, \hat{\boldsymbol{o}})$:

$$L_G^{\mathrm{GAN}}(\boldsymbol{o}, \hat{\boldsymbol{o}}) = L(\boldsymbol{o}, \hat{\boldsymbol{o}}) + \omega L_{D,1}^{\mathrm{GAN}}(\hat{\boldsymbol{o}}), \tag{7}$$

where $L_{D,1}^{\mathrm{GAN}}(\hat{\boldsymbol{o}})$ means adversarial loss to deceive a discriminator. The balance between the two loss functions $L(\boldsymbol{o}, \hat{\boldsymbol{o}})$ and $L_{D,1}^{\mathrm{GAN}}(\hat{\boldsymbol{o}})$ is controlled by the GAN weight ω. This framework is helpful in minimizing the divergence between the natural and generated speech parameters. In the synthesis phase, acoustic features are predicted by the trained generator using forward propagation and smooth speech parameters trajectories are generated by the MLPG algorithm in the same fashion as the conventional DNN-based system.

3.2. Conditional GAN-based singing voice synthesis

Although the GAN-based framework can minimize the adversarial loss, the discriminator may not be able to capture context-

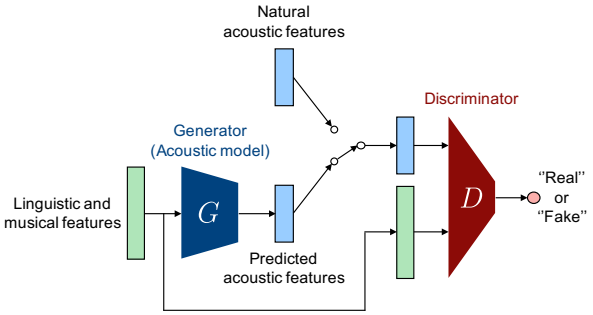

Fig. 3. The proposed CGAN-based training framework for singing voice synthesis.

dependent differences between natural and generated acoustic features. Hence, we introduced a conditional generative adversarial network (CGAN) [18] framework to discriminate the acoustic feature more appropriately. Figure 3 gives an overview of the CGAN-based training framework for singing voice synthesis. The input of the discriminator is a joint vector of an acoustic feature vector and score feature vector representing linguistic and musical features, so the discriminator can distinguish natural or generated features considering linguistic and musical information. The loss functions of the discriminator and the generator, $L_D^{\mathrm{CGAN}}(\boldsymbol{o}, \hat{\boldsymbol{o}})$ and $L_G^{\mathrm{CGAN}}(\boldsymbol{o}, \hat{\boldsymbol{o}})$ can be defined as

$$L_D^{\mathrm{CGAN}}(\boldsymbol{o}, \hat{\boldsymbol{o}}) = L_{D,1}^{\mathrm{CGAN}}(\boldsymbol{o}, \boldsymbol{l}) + L_{D,0}^{\mathrm{CGAN}}(\hat{\boldsymbol{o}}, \boldsymbol{l}), \tag{8}$$

$$L_G^{\mathrm{CGAN}}(\boldsymbol{o}, \hat{\boldsymbol{o}}) = L(\boldsymbol{o}, \hat{\boldsymbol{o}}) + \omega L_{D,1}^{\mathrm{CGAN}}(\hat{\boldsymbol{o}}, \boldsymbol{l}), \tag{9}$$

where $L_{D,1}^{\mathrm{CGAN}}(\boldsymbol{o}, \boldsymbol{l})$ and $L_{D,0}^{\mathrm{CGAN}}(\hat{\boldsymbol{o}}, \boldsymbol{l})$ are given by

$$L_{D,1}^{\mathrm{CGAN}}(\boldsymbol{o}, \boldsymbol{l}) = -\frac{1}{T}\sum_{t=1}^{T} \log D(\boldsymbol{o}_t, \boldsymbol{l}_t), \tag{10}$$

$$L_{D,0}^{\mathrm{CGAN}}(\hat{\boldsymbol{o}}, \boldsymbol{l}) = -\frac{1}{T}\sum_{t=1}^{T} \log\big(1 - D(\hat{\boldsymbol{o}}_t, \boldsymbol{l}_t)\big). \tag{11}$$

In this framework, the generator and the discriminator are trained to minimize $L_G^{\mathrm{CGAN}}(\boldsymbol{o}, \hat{\boldsymbol{o}})$ in Eq. (9) and $L_D^{\mathrm{CGAN}}(\boldsymbol{o}, \hat{\boldsymbol{o}})$ in Eq. (8), respectively.

4. EXPERIMENTS

4.1. Experimental conditions

In this experiment, 70 Japanese children's songs (total: 70 min) by female singer f001 were used. For training, 60 songs were used, and the others were used for testing. Singing voice signals were sampled at 48 kHz and windowed with a 5-ms shift. The feature vectors consisted of 0-th through 49-th STRAIGHT mel-cepstral coefficients, log F_0 value, 0-th through 24-th mel-cepstral analysis aperiodicity measures, and 2-dimensional vibrato parameters. Mel-cepstral coefficients were extracted by STRAIGHT [20]. The vibrato parameter vectors consisted of amplitude (cent) and frequency (Hz).

Five-state, left-to-right, no-skip HSMMs were used to obtain time alignment of score features to acoustic features for training the DNN-based acoustic models. The decision tree-based context clustering technique was separately applied to distributions for the spectrum, excitation, state duration, and time-lag. The spectrum stream was modeled with single multivariate Gaussian distributions. The excitation stream was modeled with multi-space probability distributions HSMMs (MSD-HSMMs) [21], each of which consisted of a Gaussian distribution for "voiced" frames and a discrete distribution for "unvoiced" frames. The vibrato stream was also modeled with MSD-HSMMs, each of which consisted of a Gaussian distribution for "vibrato" frames and a discrete distribution for "no-vibrato" frames. The MDL criterion [22] was used to control the size of the decision trees.

4.2. Experiment 1

In this experiment, the following three systems were compared.

- **Baseline**: Conventional DNN-based system trained by minimizing the loss function in Eq. (1)
- **GAN-mgc**: Proposed system using GAN-based framework trained by minimizing the loss function in Eq. (7)
- **CGAN-mgc**: Proposed system using CGAN-based framework trained by minimizing the loss function in Eq. (9)

The input feature vector for the above three systems was an 842-dimensional feature vector consisting of 734 binary features for categorical linguistic contexts, 108 numerical features for numerical contexts, and duration features including the duration of the current phoneme and the position of the current frame. In the DNN of **Baseline** and the generator of **GAN-mgc** and **CGAN-mgc**, the output feature vector was a 236-dimensional feature vector consisting of 50 mel-cepstral coefficients, log F_0 value, 25 dimensional mel-cepstral analysis aperiodicity measures, 2-dimensional vibrato parameters and their dynamic features (delta and delta-delta), a voiced/unvoiced binary value, and a vibrato/no-vibrato binary value. The discriminator of **GAN-mgc** used the 150-dimensional feature vector consisting of 50 mel-cepstral coefficients and their dynamic features (delta and delta-delta), and the one of **CGAN-mgc** used the joint vector of the 150-dimensional feature vector, which is the same as the input vector in **GAN-mgc** and the 842-dimensional feature vector, which is the score feature vector as the generator input vector. **GAN-mgc** and **CGAN-mgc** output a single scalar representing the probability. As the generator of GAN-mgc and CGAN-mgc, a single network that modeled every spectral, excitation, aperiodicity, and vibrato parameter was trained.

The architecture of the DNN-based acoustic models was a 3-hidden-layer feed-forward neural network with 2048 units per layer. The architecture of the discriminator was a 2-hidden-layer feed-forward neural network with 1024 units per layer. The sigmoid and linear activation functions were used for the hidden and output layers of the generator, respectively, and the leaky ReLU and the sigmoid activation functions were used for the hidden and output layers of the discriminator, respectively. The weight parameters of all neural networks were initialized randomly. The weight parameters of the DNN in **Baseline** and the discriminator in **GAN-mgc** and **CGAN-mgc** were initialized randomly. The DNNs were optimized by minimizing the loss function in Eq. (1). In **GAN-mgc** and **CGAN-mgc**, the weight parameters of the generator were initialized by a trained DNN in **Baseline**, and then they were optimized by minimizing the loss function in Eq. (4) or Eq. (8). Finally, the generator and discriminator were both trained simultaneously using the adversarial training framework described in Section 3.

To objectively evaluate the performance of the systems, we calculated the averaged global variance (GV) of the mel-cepstral coefficients. Figure 4 shows the averages from the evaluation data. As

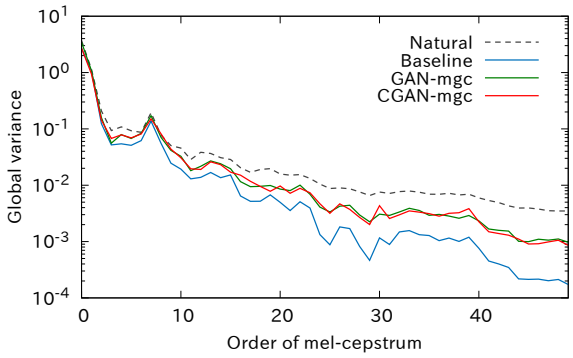

Fig. 4. Averaged GVs of mel-cepstral coefficients.

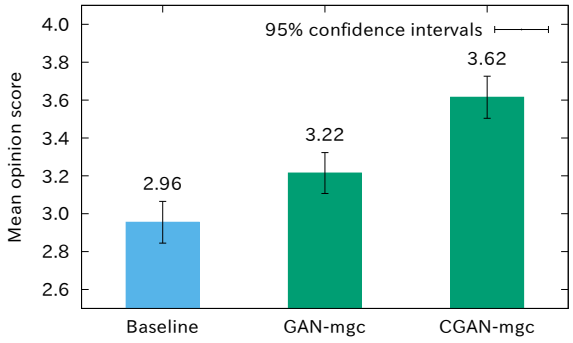

Fig. 5. Results of MOS test comparing GAN-based systems with CGAN-based systems

shown in the figure, **GAN-mgc** and **CGAN-mgc** improved the averaged GV compared to the **Baseline**. This result indicates that the over-smoothing problem is alleviated by using the adversarial training method.

The naturalness of the synthesized singing voice was assessed by the mean opinion score (MOS) test method. The subjects were ten Japanese students in our research group. Twenty musical phrases were chosen at random from the test songs. In the MOS test, after listening to each test sample, the subjects were asked to assign the sample a five-point naturalness score (5: natural – 1: poor).

Figure 5 shows the results of subjective evaluation scores. The GAN-based systems, **GAN-mgc** and **CGAN-mgc**, outperformed **Baseline** significantly. These results clearly show that the naturalness of synthesized singing voices was improved by introducing adversarial training. Also, comparing **GAN-mgc** and **CGAN-mgc**, **CGAN-mgc** obtained a better score than **GAN-mgc**. This result suggests that using the discriminator conditioned by linguistic and musical information is effective for the adversarial training for singing voice synthesis.

4.3. Experiment 2

In this experiment, the following three systems were compared to evaluate the effectiveness of the acoustic features input to the discriminator for the adversarial training.

- **Baseline**: Conventional DNN-based system trained by minimizing the loss function in Eq. (1)
- **CGAN-mgc**: Proposed CGAN-based system that used spectral features as the input of the discriminator
- **CGAN-all**: Proposed CGAN-based system that used spectral, excitation, aperiodicity, and vibrato parameters as the input of the discriminator

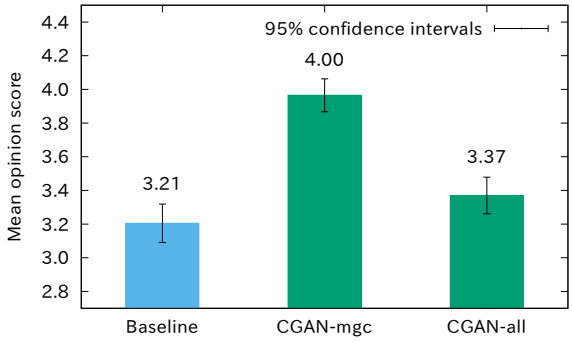

Fig. 6. The results of MOS test comparing the acoustic features input to the discriminator

In **CGAN-all**, the input and output feature vectors for the generator were the same as those used in **CGAN-mgc**. The input of the discriminator in **CGAN-all** used the joint vector of the 236-dimensional feature vector, which consisted of spectral, excitation, aperiodicity, and vibrato parameters, and 842-dimensional feature vector, which was the same vector as the input feature vector of the generator. The MOS test was conducted in the same manner as experiment 1.

Figure 6 shows the results of subjective evaluation scores. Comparing **Baseline** to **CGAN-all**, **CGAN-all** outperformed **Baseline**, though comparing **CGAN-all** to **CGAN-mgc**, **CGAN-all** did not reach **CGAN-mgc**. This result suggests that the excitation, aperiodicity, and vibrato parameters were not effective in the proposed CGAN-based method. It seems that this is because the discriminator classifies input feature vectors as real or fake frame-by-frame. It is expected that the CGAN-based method can model acoustic features (e.g., the excitation parameters) more effectively by classifying acoustic features as a sequence in the discriminator.

5. CONCLUSIONS

In this paper, we proposed the DNN-based singing voice synthesis using generative adversarial networks (GANs). The proposed method can model acoustic features more accurately than conventional DNN-based systems, and it can solve the over-smoothing problem. Experimental results show that the proposed method can alleviate the over-smoothing problem and improve the naturalness of a synthesized singing voice compared to a conventional DNN-based system. Future work includes exploring a sequential model structure for the discriminator to distinguish acoustic features, including $F0$ sequences.

6. ACKNOWLEDGMENTS

This research was partly funded by JSPS KAKENHI Grant Number JP18K11163, CASIO SCIENCE PROMOTION FOUNDATION, and Microsoft Development Co., Ltd.

7. REFERENCES

[1] M. Nishimura, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "Singing voice synthesis based on deep neural networks," *Proceedings of Interspeech 2016*, pp. 2478–2482, 2016.

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Proceedings of NIPS 2014*, pp. 2672–2680, 2014.

[3] Y. Saito, S. Takamichi, and H. Saruwatari, "Statistical parametric speech synthesis incorporating generative adversarial networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 1, pp. 84–96, 2018.

[4] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[6] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[7] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," *Proceedings of ICASSP 2013*, pp. 7962–7966, 2013.

[8] Y. Qian, Y. Fan, W. Hu, and F. K. Soong, "On the training aspects of deep neural network (DNN) for parametric TTS synthesis," *Proceedings of ICASSP 2014*, pp. 3829–3833, 2014.

[9] O. Watts, G. E. Henter, T. Merritt, Z. Wu, and S. King, "From HMMs to DNNs: where do the improvements come from?," *Proceedings of ICASSP 2016*, pp. 5505–5509, 2016.

[10] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[11] S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. Courville, and Y. Bengio, "SampleRNN: An unconditional end-to-end neural audio generation model," *arXiv preprint arXiv:1612.07837*, 2016.

[12] Z. Jin, A. Finkelstein, G. J. Mysore, and J. Lu, "FFTnet: A real-time speaker-dependent neural vocoder," *Proceedings of ICASSP 2018*, pp. 2251–2255, 2018.

[13] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," *Proceedings of ICML 2016*, vol. 48, pp. 1060–1069, 20–22 Jun 2016.

[14] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[15] S. Pascual, A. Bonafonte, and J. Serra, "SEGAN: Speech enhancement generative adversarial network," *arXiv preprint arXiv:1703.09452*, 2017.

[16] L.-C. Yang, S.-Y. Chou, and Y.-H. Yang, "MidiNet: A convolutional generative adversarial network for symbolic-domain music generation," *arXiv preprint arXiv:1703.10847*, 2017.

[17] S. Yang, L. Xie, X. Chen, X. Lou, X. Zhu, D. Huang, and H. Li, "Statistical parametric speech synthesis using generative adversarial networks under a multi-task learning framework," *Proceedings of ASRU 2017*, pp. 685–691, 2017.

[18] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1406.2661*, 2014.

[19] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," *Proceedings of ICASSP 2000*, vol. 3, pp. 1315–1318, 2000.

[20] H. Kawahara, I. Masuda-Katsuse, and A. 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.

[21] K. Tokuda, T. Masuko, N. Miyazaki, and T. Kobayashi, "Hidden Markov models based on multi-space probability distribution for pitch pattern modeling," *Processings of ICASSP 1999*, vol. 1, pp. 229–232, 1999.

[22] K. Shinoda and T. Watanabe, "Acoustic modeling based on the MDL principle for speech recognition," *Proceedings of Eurospeech 1997*, pp. 99–102, 1997.