Statistical Parametric Speech Synthesis Incorporating Generative Adversarial Networks

Yuki Saito, NonMember, Shinnosuke Takamichi, Member, IEEE and Hiroshi Saruwatari, Member, IEEE

Abstract—A method for statistical parametric speech synthesis incorporating generative adversarial networks (GANs) is proposed. Although powerful deep neural networks (DNNs) techniques can be applied to artificially synthesize speech waveform, the synthetic speech quality is low compared with that of natural speech. One of the issues causing the quality degradation is an over-smoothing effect often observed in the generated speech parameters. A GAN introduced in this paper consists of two neural networks: a discriminator to distinguish natural and generated samples, and a generator to deceive the discriminator. In the proposed framework incorporating the GANs, the discriminator is trained to distinguish natural and generated speech parameters, while the acoustic models are trained to minimize the weighted sum of the conventional minimum generation error training algorithm regardless its hyper-parameter settings. Furthermore, we investigated the effect of the divergence of various GANs, and found that a Wasserstein GAN minimizing the Earth-Mover’s distance works the best in terms of improving synthetic speech quality.

Index Terms—Statistical parametric speech synthesis, text-to-speech synthesis, voice conversion, deep neural networks, generative adversarial networks, over-smoothing.

I. INTRODUCTION

STATISTICAL parametric speech synthesis (SPSS) [1] is a technique that aims to generate natural-sounding synthetic speech. Text-to-speech (TTS) synthesis [2] is a technique for synthesizing speech from text, and voice conversion (VC) [3] is a technique for synthesizing speech from another one while preserving linguistic information of original speech. In SPSS, acoustic models represent the relationship between input features and acoustic features. Recently, deep neural networks (DNNs) [4] have been utilized as the acoustic models for TTS and VC because they can model the relationship between input features and acoustic features more accurately than conventional hidden Markov models [5] and Gaussian mixture models [6]. These acoustic models are trained with several training algorithms such as the minimum generation error (MGE) criterion [7], [8]. Techniques for training the acoustic models to generate high-quality speech are widely studied since they can be used for both TTS and VC. However, the speech parameters generated from these models tend to be over-smoothed, and the resultant quality of speech is still low compared with that of natural speech [1], [9]. The over-smoothing effect is a common issue in both TTS and VC.

One way to improve speech quality is to reduce the difference between natural and generated speech parameters. For instance, since the parameter distributions of natural and synthetic speech are significantly different [10], we can improve the synthetic speech quality by transforming the generated speech parameters so that their distribution is close to that of natural speech. This can be done by, for example, modeling the probability distributions in a parametric [6] or non-parametric [11] way in the training stage, and then, generating or transforming the synthetic speech parameters by using the distributions. The more effective approach is to use analytically derived features correlated to the quality degradation of the synthetic speech. Global variance (GV) [6] and modulation spectrum (MS) [12] are well-known examples for reproducing natural statistics. These features work as a constraint in the training/synthesis stage [13], [14]. Nose and Ito [15] and Takamichi et al. [13] proposed methods that reduce the difference between the Gaussian distributions of natural and generated GV and MS. However, quality degradation is still a critical problem.

In order to address this quality problem, in this paper we propose a novel method using generative adversarial networks (GANs) for training acoustic models in SPSS. A GAN consists of two neural networks: a discriminator to distinguish natural and generated samples, and a generator to deceive the discriminator. Based on the framework, we define a new training criterion for the acoustic models; the criterion is the weighted sum of the conventional MGE training and an adversarial loss. The adversarial loss makes the discriminator recognize the generated speech parameters as natural. Since the objective of the GANs is to minimize the divergence (i.e., the distribution difference) between the natural and generated speech parameters, our method effectively alleviates the effect of over-smoothing the generated speech parameters. Moreover, our method can be regarded as a generalization of the conventional method using explicit modeling of analytically derived features such as GV and MS because it effectively minimizes the divergence without explicit statistical modeling. Also, the discriminator used in our method can be interpreted as anti-spoofing, namely, a technique for detecting synthetic speech and preventing voice spoofing attack. Accordingly, techniques and ideas concerning anti-spoofing can be applied.
to the training. We evaluated the effectiveness of the proposed method in DNN-based TTS and VC, and found that the proposed algorithm generates more natural spectral parameters and \( F_0 \) than those of the conventional MGE training algorithm and improves the synthetic speech quality regardless its hyper-parameter settings which control the weight of the adversarial loss. Furthermore, we investigated the effect of the divergence of various GANs, including image-processing-related ones such as the least squares GAN (LS-GAN) and the Wasserstein GAN (W-GAN), and speech-processing-related ones such as the \( f \)-divergence GAN (\( f \)-GAN). The results of the investigation demonstrate that the W-GAN minimizing the Earth-Mover’s distance works the best in regard to improving synthetic speech quality.

In Section II of this paper, we briefly review conventional training algorithms in DNN-based TTS and VC. Section III introduces GANs and proposes a method for speech synthesis incorporating those GANs. Section IV presents the experimental evaluations. We conclude in Section V with a summary.

II. CONVENTIONAL DNN-BASED SPSS

This section describes the conventional training algorithm for DNN-based SPSS, including TTS and VC.

A. DNN-based TTS

1) DNNs as Acoustic Models: In DNN-based TTS [16], acoustic models representing the relationship between linguistic features and speech parameters consist of layered hierarchical networks. In training the models, we minimize the loss function calculated using the speech parameters of natural and synthetic speech. Let \( x = [x_1^T, \cdots, x_t^T, \cdots, x_T^T]^T \) be a linguistic feature sequence, \( y = [y_1^T, \cdots, y_t^T, \cdots, y_T^T]^T \) be a natural speech parameter sequence, and \( \hat{y} = [\hat{y}_1^T, \cdots, \hat{y}_t^T, \cdots, \hat{y}_T^T]^T \) be a generated speech parameter sequence, where \( t \) and \( T \) denote the frame index and total frame length, respectively. \( x_t \) and \( y_t = [y_t(1), \cdots, y_t(D)]^T \) are a linguistic parameter vector and a \( D \)-dimensional speech parameter vector at frame \( t \), respectively.

2) Acoustic model training: The DNNs that predict a natural static-dynamic speech feature sequence \( Y = [Y_1^T, \cdots, Y_t^T, \cdots, Y_T^T]^T \) from \( x \) are trained to minimize a defined training criterion. \( Y_t = [y_t^T, \Delta y_t^T, \Delta \Delta y_t^T]^T \) is a natural static-dynamic speech feature at frame \( t \). Given a predicted static-dynamic speech feature sequence \( \hat{Y} = [\hat{Y}_1^T, \cdots, \hat{Y}_t^T, \cdots, \hat{Y}_T^T]^T \), the most standard criterion is the mean squared error (MSE) \( L_{MSE}(Y, \hat{Y}) \) between \( Y \) and \( \hat{Y} \) defined as follows:

\[
L_{MSE}(Y, \hat{Y}) = \frac{1}{T} (\hat{Y} - Y)^T (\hat{Y} - Y). \tag{1}
\]

A set of the model parameters \( \theta_G \) (e.g., weight and bias of DNNs) is updated by the backpropagation algorithm using the gradient \( \nabla_{\theta_G} L_{MSE}(Y, \hat{Y}) \).

To take the static-dynamic constraint into account, the minimum generation error (MGE) training algorithm was proposed [8]. In MGE training, the loss function \( L_{MGE}(y, \hat{y}) \) is defined as the mean squared error between natural and generated speech parameters as follows:

\[
L_{MGE}(y, \hat{y}) = \frac{1}{T} (\hat{y} - y)^T (\hat{y} - y) = \frac{1}{T} (R\hat{Y} - y)^T (R\hat{Y} - y). \tag{2}
\]

\( R \) is a DT-by-3DT matrix given as

\[
R = (W^T \Sigma^{-1} W)^{-1} W^T \Sigma^{-1}, \tag{3}
\]

where \( W \) is a 3DT-by-3DT matrix for calculating dynamic features [5] and \( \Sigma = \text{diag}([\Sigma_1, \cdots, \Sigma_t, \cdots, \Sigma_T]) \) is a 3DT-by-3DT covariance matrix, where \( \Sigma_t \) is a 3D-by-3D covariance matrix at frame \( t \). \( \Sigma \) is separately estimated using training data. We define the speech parameter prediction as \( \hat{y} = R\hat{Y} = G(x; \theta_G) \), where \( \theta_G \) denotes the acoustic model parameters and it is updated by the backpropagation algorithm using the gradient of the generation error, \( \nabla_{\theta_G} L_{MGE}(y, \hat{y}) \).

As described in [8], the gradient includes \( \nabla_{\hat{Y}} L_{MGE}(y, \hat{y}) \) given as \( R^T (\hat{y} - y)/T \).

Phoneme duration is predicted in the same manner without dynamic feature calculation. Let \( d = [d_1, \cdots, d_p, \cdots, d_P]^T \) be a natural phoneme duration sequence, and \( \hat{d} = [\hat{d}_1, \cdots, \hat{d}_p, \cdots, \hat{d}_P]^T \) be a duration sequence generated using duration models described as DNNs. \( p \) is the phoneme index and \( P \) is the total number of phonemes. The model parameters are updated to minimize \( L_{MSE}(d, \hat{d}) \).

B. DNN-based VC

DNN-based acoustic models for VC convert input speech features to desired output speech features. In training, a dynamic time warping algorithm is used to temporally align source and target speech features. Using the aligned features, \( x \) and \( y \), the acoustic models are trained to minimize \( L_{MGE}(y, \hat{y}) \), the same as DNN-based TTS.

III. DNN-BASED SPSS INCORPORATING GAN

A. Generative Adversarial Networks (GANs) [17]

A GAN is a framework for learning deep generative models, which simultaneously trains two DNNs: a generator and discriminator \( D(y; \theta_D) \). \( \theta_D \) is a set of the model parameters of the discriminator. The value obtained by taking the sigmoid function from the discriminator’s output, \( 1/(1+\exp(-D(y))) \), represents the posterior probability that input \( y \) is natural data. The discriminator is trained to make the posterior probability 1 for natural data and 0 for generated data, while the generator is trained to deceive the discriminator; that is, it tries to make the discriminator make the posterior probability 1 for generated data.

In the GAN training, the two DNNs are iteratively updated by minibatch stochastic gradient descent. First, by using natural data \( y \) and generated data \( \hat{y} \), we calculate the discriminator
loss $L_{DG}^{(GAN)}(y, \hat{y})$ defined as the following cross-entropy function:

$$L_{DG}^{(GAN)}(y, \hat{y}) = -\frac{1}{T} \sum_{t=1}^{T} \log \frac{1}{1 + \exp (-D(y_t))}$$

$$- \frac{1}{T} \sum_{t=1}^{T} \log \left(1 - \frac{1}{1 + \exp (-D(\hat{y}_t))}\right).$$

(4)

$\theta_D$ is updated by using the stochastic gradient $\nabla_{\theta_D} L_{DG}^{(GAN)}(y, \hat{y})$. Figure [1] illustrates the procedure for computing the discriminator loss. After updating the discriminator, we calculate the adversarial loss of the generator $L_{ADV}^{(GAN)}(\hat{y})$ which deceives the discriminator as follows:

$$L_{ADV}^{(GAN)}(\hat{y}) = -\log \frac{1}{1 + \exp (-D(\hat{y}))}.$$ 

(5)

A set of the model parameters of the generator $\theta_G$ is updated by using the stochastic gradient $\nabla_{\theta_G} L_{ADV}^{(GAN)}(\hat{y})$. Goodfellow et al. [17] showed this adversarial framework minimizes the approximated Jensen–Shannon (JS) divergence between two distributions of natural and generated data.

### B. Acoustic Model Training Incorporating GAN

Here, we describe a novel training algorithm for SPSS which incorporates the GAN. As for the proposed algorithm, acoustic models are trained to deceive the discriminator that distinguishes natural and generated speech parameters.

The loss function of speech synthesis is defined as the following:

$$L_G(y, \hat{y}) = L_{MGE}(y, \hat{y}) + \omega_D \frac{E_{L_{MGE}}}{E_{L_{ADV}}} L_{ADV}^{(GAN)}(\hat{y}),$$

(6)

where $L_{ADV}^{(GAN)}(\hat{y})$ makes the discriminator recognize the generated speech parameters as natural, and minimizes the divergence between the distributions of the natural and generated speech parameters. Therefore, the proposed loss function not only minimizes the generation error but also makes the distribution of the generated speech parameters close to that of natural speech. $E_{L_{MGE}}$ and $E_{L_{ADV}}$ denote the expectation values of $L_{MGE}(y, \hat{y})$ and $L_{ADV}^{(GAN)}(\hat{y})$, respectively. Their ratio $E_{L_{MGE}}/E_{L_{ADV}}$ is the scale normalization term between the two loss functions, and the hyper-parameter $\omega_D$ controls the weight of the second term. When $\omega_D = 0$, the loss function is equivalent to the conventional MGE training, and when $\omega_D = 1$, the two loss functions have equal weights. A set of the model parameters of the acoustic models $\theta_G$ is updated by using the stochastic gradient $\nabla_{\theta_G} L_G(y, \hat{y})$. Figure [2] illustrates the procedure for computing the proposed loss function. In our algorithm, the acoustic models and discriminator are iteratively optimized, as shown in Algorithm 1. When one module is being updated, the model parameters of the another are fixed; that is, although the discriminator is included in the forward path to calculate $L_{ADV}^{(GAN)}(\hat{y})$ in $L_G(y, \hat{y})$, $\theta_D$ is not updated by the backpropagation for the acoustic models.

The discriminator used in our method can be regarded as a DNN-based anti-spoofing (voice spoofing detection) [18], [19] that distinguishes natural and synthetic speech. From this

---

**Algorithm 1 Iterative optimization for acoustic models and discriminator**

1: $\eta :=$ learning rate
2: for number of training iterations do
3: for all training data $(x, y)$ do
4: generate $\hat{y}$ from the acoustic models:
   $$\hat{y} = G(x).$$
5: update $\theta_D$ while fixing $\theta_G$:
   $$\theta_D \leftarrow \theta_D - \eta \nabla_{\theta_D} L_{DG}^{(GAN)}(y, \hat{y}).$$
6: update $\theta_G$ while fixing $\theta_D$:
   $$\theta_G \leftarrow \theta_G - \eta \nabla_{\theta_G} L_G(y, \hat{y}).$$
7: end for
8: end for
minimizes the cross-entropy function by using the isochrony-level duration, while the generator minimizes the weighted sum of the MSE between natural and generated phoneme durations and the adversarial loss using the isochrony-level durations. Since the calculation of the isochrony-level duration is represented as the matrix multiplication shown in Fig. 4, the backpropagation is done using the transpose of the transformation matrix.

D. GANs to Be Applied to The Proposed Method

The GAN framework works as a divergence minimization between natural and generated speech parameters. As described in Section III-B, the original GAN minimizes the approximated JS divergence. From the perspective of the divergence minimization, we further introduce additional GANs minimizing other divergences: f-GAN, Wasserstein GAN (W-GAN), and least squares GAN (LS-GAN). The divergence of the f-GAN is strongly related to speech processing such as a nonnegative matrix factorization and the effectiveness of the W-GAN and LS-GAN in the image processing is known. The discriminator loss \( L^{(\text{f-GAN})}_D (\hat{y}; \hat{y}) \) and adversarial loss \( L^{(\text{ADV})}_D (\hat{y}) \) introduced below can be used instead of Eqs. (4) and (5), respectively.

1) f-GAN: The f-GAN is the unified framework that encompasses the original GAN. The difference between distributions of natural and generated data is defined as the f-divergence, which is a large class of different divergences including the Kullback–Leibler (KL) and JS divergence. The f-divergence \( D_f (y || \hat{y}) \) is defined as follows:

\[
D_f (y || \hat{y}) = \int q (\hat{y}) f \left( \frac{p(y)}{q(\hat{y})} \right) d\hat{y},
\]

where \( p(\cdot) \) and \( q(\cdot) \) are absolutely continuous density functions of \( y \) and \( \hat{y} \), respectively. \( f(\cdot) \) is a convex function satisfying \( f(1) = 0 \). Although various choices of \( f(\cdot) \) for recovering popular divergences are available, we adopt ones related to speech processing.

KL-GAN: Defining \( f(r) = r \log r \) gives the KL divergence as follows:

\[
D_{\text{KL}} (y || \hat{y}) = \int p(y) \log \frac{p(y)}{q(\hat{y})} dy.
\]

The discriminator loss \( L^{(\text{KL-GAN})}_D (y; \hat{y}) \) is defined as follows:

\[
L^{(\text{KL-GAN})}_D (y; \hat{y}) = - \frac{1}{T} \sum_{t=1}^{T} D(y_t)
+ \frac{1}{T} \sum_{t=1}^{T} \exp (D(\hat{y}_t) - 1),
\]

while the adversarial loss \( L^{(\text{KL-GAN})}_{\text{ADV}} (\hat{y}) \) is defined as follows:

\[
L^{(\text{KL-GAN})}_{\text{ADV}} (\hat{y}) = - \frac{1}{T} \sum_{t=1}^{T} D(\hat{y}_t).
\]

Reversed KL (RKL)-GAN: Since the KL divergence is not symmetric, the reversed version, called reversed KL (RKL)}
divergence \( D_{RKL}(y||\hat{y}) \) differs from \( D_{KL}(y||\hat{y}) \), which is defined as follows:
\[
D_{RKL}(y||\hat{y}) = \int q(\hat{y}) \log \frac{q(\hat{y})}{p(y)} dy = D_{KL}(\hat{y}||y). \tag{11}
\]
Defining \( f(r) = -\log r \) gives the discriminator loss \( L_{D}(RKL\text{-GAN})(y, \hat{y}) \) as follows:
\[
L_{D}(RKL\text{-GAN})(y, \hat{y}) = \frac{1}{T} \sum_{t=1}^{T} \exp \left(-D(y_t)\right)
+ \frac{1}{T} \sum_{t=1}^{T} (-1 + D(\hat{y}_t)), \tag{12}
\]
while the adversarial loss \( L_{ADV}(RKL\text{-GAN})(\hat{y}) \) is defined as follows:
\[
L_{ADV}(RKL\text{-GAN})(\hat{y}) = \frac{1}{T} \sum_{t=1}^{T} \exp \left(-D(\hat{y}_t)\right). \tag{13}
\]

JS-GAN: The JS divergence without approximation can be formed within the \( f \)-GAN framework. Defining \( f(r) = -(r+1) \log \frac{r}{c} + r \log r \) gives the JS divergence as follows:
\[
D_{JS}(y||\hat{y}) = \frac{1}{2} \int p(y) \log \frac{2p(y)}{p(y) + q(\hat{y})} dy
+ \frac{1}{2} \int q(\hat{y}) \log \frac{2q(\hat{y})}{p(y) + q(\hat{y})} dy. \tag{14}
\]
the discriminator loss \( L_{D}(JS\text{-GAN})(y, \hat{y}) \) is defined as follows:
\[
L_{D}(JS\text{-GAN})(y, \hat{y}) = -\frac{1}{T} \sum_{t=1}^{T} \log \left(1 + \exp \left(-D(y_t)\right)\right)
- \frac{1}{T} \sum_{t=1}^{T} \log \left(2 - \frac{2}{1 + \exp \left(-D(\hat{y}_t)\right)}\right), \tag{15}
\]
while the adversarial loss \( L_{ADV}(JS\text{-GAN})(\hat{y}) \) is defined as follows:
\[
L_{ADV}(JS\text{-GAN})(\hat{y}) = -\frac{1}{T} \sum_{t=1}^{T} \log \left(2 - \frac{2}{1 + \exp \left(-D(\hat{y}_t)\right)}\right). \tag{16}
\]
Note that, the approximated JS divergence minimized by the original GAN is \( 2D_{JS}(y||\hat{y}) - \log(4) \) \( \tag{17} \).

2) Wasserstein GAN (W-GAN) \[23\]: To stabilize the extremely unstable training of the original GAN, Arjovsky et al. \[23\] proposed the W-GAN, which minimizes the Earth-Mover’s distance (Wasserstein-1). The Earth-Mover’s distance is defined as follows:
\[
D_{EM}(y, \hat{y}) = \inf_{\gamma} \mathbb{E}_{(y, \hat{y}) \sim \gamma} \left[ ||y - \hat{y}|| \right], \tag{17}
\]
where \( \gamma(y, \hat{y}) \) is the joint distribution whose marginals are respectively the distributions of \( y \) and \( \hat{y} \). On the basis of the Kantorovich–Rubinstein duality \[28\], the discriminator loss \( L_{D}(W\text{-GAN})(y, \hat{y}) \) is defined as follows:
\[
L_{D}(W\text{-GAN})(y, \hat{y}) = -\frac{1}{T} \sum_{t=1}^{T} D(y_t) + \frac{1}{T} \sum_{t=1}^{T} D(\hat{y}_t), \tag{18}
\]
while the adversarial loss \( L_{ADV}(W\text{-GAN})(\hat{y}) \) is defined as follows:
\[
L_{ADV}(W\text{-GAN})(\hat{y}) = -\frac{1}{T} \sum_{t=1}^{T} D(\hat{y}_t). \tag{19}
\]
We assume the discriminator to be the \( K \)-Lipschitz function. Namely, after updating the discriminator, we clamp its weight parameters to a fixed interval such as \([-0.01, 0.01]\).

3) Least Squares GAN (LS-GAN) \[24\]: To avoid the gradient vanishing problem of the original GAN using the sigmoid cross entropy, Mao et al. \[24\] proposed the LS-GAN, which formulates the objective function minimizing the mean squared error. The discriminator loss \( L_{D}(LS\text{-GAN})(y, \hat{y}) \) is defined as follows:
\[
L_{D}(LS\text{-GAN})(y, \hat{y}) = \frac{1}{2T} \sum_{t=1}^{T} (D(y_t) - b)^2
+ \frac{1}{2T} \sum_{t=1}^{T} (D(\hat{y}_t) - a)^2, \tag{20}
\]
while the adversarial loss \( L_{ADV}(LS\text{-GAN})(\hat{y}) \) is defined as follows:
\[
L_{ADV}(LS\text{-GAN})(\hat{y}) = \frac{1}{2T} \sum_{t=1}^{T} (D(\hat{y}_t) - c)^2, \tag{21}
\]
where \( a, b, \) and \( c \) denote the labels that make the discriminator recognize the generated data as generated, the natural data as natural, and the generated data as natural. When they satisfy the conditions \( b - c = 1 \) and \( b - a = 2 \), the divergence to be minimized is the Pearson \( \chi^2 \) divergence between \( p(y) + q(\hat{y}) \) and \( 2q(\hat{y}) \). Because we found that these conditions degrade quality of synthetic speech, we used alternative conditions suggested in Eq. (9) of \[24\], i.e., \( a = 0, b = 1, \) and \( c = 1 \).

E. Discussions

The proposed loss function (Eq. 6) is the combination of a multi-task learning algorithm using discriminators \[29\] and GANs. In defining \( L_G(y, \hat{y}) = L_{ADV}(\hat{y}) \), the loss function is equivalent to that for the GAN. Comparing with the GANs, our method is a fully supervised setting, i.e., we utilize the referred input and output parameters \[30\] without a latent variable. Also, since only the backpropagation algorithm is used for training, a variety of DNN architectures such as long short-term memory (LSTM) \[31\] can be used as the acoustic models and discriminator.

Using the designed feature function \( \phi(\cdot) \), we can choose not only analytically derived features (e.g., GV and MS) but also automatically derived features (e.g., auto-encoded features \[32\]).

As described above, our algorithm makes the distribution of the generated speech parameters close to that of the natural speech. Since we perform generative adversarial training with DNNs, our algorithm comes to have a more complicated probability distribution than the conventional Gaussian distribution. Figure 5 plots natural and generated speech parameters with several mel-cepstral coefficient pairs. Whereas the parameters of the conventional algorithm are narrowly distributed, those of the proposed algorithm are as widely distributed as the natural
Mel-cepstral coefficients were extracted from one utterance of the evaluation data, a verification of the proposed algorithm, and the conventional MGE algorithm. Figure 5 illustrates scatter plots for several pairs of mel-cepstral coefficients. From the left, the figures correspond to natural speech, the conventional MGE algorithm, and the proposed algorithm (ω₀ = 1.0). These mel-cepstral coefficients were extracted from one utterance of the evaluation data.

Figure 6 plots the averaged GVs of natural and generated speech parameters. The MIC ranges from 0.0 to 1.0, and the two variables with a strong correlation have a value closer to 1.0. From the left, the figures correspond to natural speech, the conventional MGE algorithm, and the proposed algorithm (ω₀ = 1.0). These MICs were calculated from one utterance of the evaluation data.

![Fig. 6. Averaged GVs of mel-cepstral coefficients.](image)

Fig. 5. Scatter plots of mel-cepstral coefficients with several pairs of dimensions. From the left, the figures correspond to natural speech, the conventional MGE algorithm, and the proposed algorithm (ω₀ = 1.0). These mel-cepstral coefficients were extracted from one utterance of the evaluation data.

Fig. 6. Averaged GVs of mel-cepstral coefficients. Dashed, black, and blue lines correspond to natural speech, the conventional MGE, and the proposed algorithm, respectively.

Speech. Moreover, we can see that the proposed algorithm has a greater effect on the distribution of the higher order of the mel-cepstral coefficients.

Here, one can explore which components (e.g., analytically derived features and intuitive reasons [33, 34]) the algorithm changes. Figure 6 plots the averaged GVs of natural and generated speech parameters. We can see that the GV generated by the proposed algorithm is closer to the natural GV than that of the one produced by the conventional algorithm. This is quite natural result because compensating distribution differences is related to minimizing moments differences [34, 35]. Then, we calculated a maximal information coefficient (MIC) [36] to quantify a nonlinear correlation among the speech parameters. The results are shown in Fig. 7. As reported in [35], we can see that there are weak correlations among the natural speech parameters, whereas strong correlations are observed among those of the generated speech parameters of the MGE training. Moreover, the generated mel-cepstral coefficients of our algorithm have weaker correlations than those of the MGE training. These results suggest that the proposed algorithm compensates not only the GV of the generated speech parameters but also the correlation among the parameters. Also, the statistics of continuous F₀, phoneme duration, and mora duration are listed in Tables I, II, and III, respectively. The bold values are the closest to natural statistics in the results. In Tables I, II, and III, “Proposed (phoneme)” and “Proposed (mora)” indicate that the proposed method applies to phoneme and mora duration, respectively. We can see that the proposed method also makes the statistics closer to those of the natural speech than the conventional method. In the results concerning duration generations, “Proposed (mora)” tends to

| Mean   | Variance |
|--------|----------|
| Natural | 4.8784   | 0.076853 |
| MGE     | 4.8388   | 0.032841 |
| Proposed (ω₀ = 1.0) | 4.8410   | 0.032968 |

Table I

| Mean   | Variance |
|--------|----------|
| Natural | 16.314   | 126.20  |
| MSE     | 14.967   | 47.665  |
| Proposed (phoneme, ω₀ = 1.0) | 14.963   | 75.471  |
| Proposed (mora, ω₀ = 1.0) | 15.074   | 73.207  |

Table II

| Mean    | Variance  |
|---------|-----------|
| Natural | 25.141    | 131.93   |
| MSE     | 24.978    | 96.682   |
| Proposed (phoneme, ω₀ = 1.0) | 24.794    | 97.568   |
| Proposed (mora, ω₀ = 1.0) | 24.978    | 96.682   |

Table III
reduce the difference in the mean rather than in the variance.

Our algorithm for spectrum and $F_0$, proposed in Section III-C, compensates the joint distribution of them. Therefore, we can perform the distribution compensation considering correlations between different features. Also, compensating dimensionality differences can be applied for deceiving the discriminator. Since the time resolutions in phoneme duration and mora duration are different, our algorithm considering isochrony is related to multi-resolution GAN and hierarchical duration modeling.

Regarding related work, Kaneko et al. proposed a generative adversarial network-based post-filter for TTS. The post-filtering process has high portability because it is independent of original speech synthesis procedures, but it comes at a high computation cost and has a heavy disk footprint in synthesis. In contrast, our algorithm can directly utilize original synthesis procedures. Also, we expect that our algorithm can be extended to waveform synthesis.

### IV. Experimental Evaluation

In this section, we evaluate the effectiveness of the proposed algorithm in terms of spectral parameters, $F_0$, and duration generation in DNN-based TTS, and then evaluate spectral parameter conversion in DNN-based VC.

#### A. Experimental Conditions in TTS Evaluation

We used speech data of a male speaker taken from the ATR Japanese speech database. The speaker uttered 503 phonetically balanced sentences. We used 450 sentences (subsets A to I) for the training and 53 sentences (subset J) for the evaluation. Speech signals were sampled at a rate of 16 kHz, and the shift length was set to 5 ms. The 0th-through-24th mel-cepstral coefficients were used as spectral parameters and $F_0$ and 5 band-aperiodicity were used as excitation parameters. The STRAIGHT analysis-synthesis system was used for the parameter extraction and the waveform synthesis. To improve training accuracy, speech parameter trajectory smoothing with a 50 Hz cutoff modulation frequency was applied to the spectral parameters in the training data. In the training phase, spectral features were normalized to have zero-mean unit-variance, and 80% of the silent frames were removed from the training data in order to increase training accuracy.

The DNN architectures are listed in Table. In the spectral parameter generation (sections IV-B-1 and IV-B-2), the acoustic models predicted static-dynamic feature sequence of the mel-cepstral coefficients (75-dim.) from the 274-dimensional linguistic features frame by frame, and the discriminator used the joint vector of the frame-wise static mel-cepstral coefficients and continuous log $F_0$ (26-dim.). In the duration generation (section IV-B-3), we constructed duration models that generate phoneme duration from corresponding linguistic features (439-dim). The acoustic models were trained using MGE training.

In the training phase, we ran the training algorithm based on minimizing the MSE (Eq. (1)) for the initialization of acoustic models and then we ran the conventional MGE training with 25 iterations. Here, “iteration” means using all the training data (450 utterances) once for training. The discriminator was initialized using natural speech and synthetic speech after the MGE training. The number of iterations for the discriminator initialization was 5. The proposed training and discriminator re-training were performed with 25 iterations. The expectation values $E_{L_{MGE}}$ and $E_{L_{ADV}}$ were estimated at each iteration step.

#### B. Evaluation in TTS

1) **Objective Evaluation with Hyper-parameter Settings:** In order to evaluate our algorithm, we calculated the parameter generation loss defined in Eq. (2) and the spoofing rate of the synthetic speech. The spoofing rate is the number of spoofing speech parameters divided by the total number of synthetic speech parameters in the evaluation data. Here, “spoofing synthetic speech parameter” indicates a parameter for which the discriminator recognized the synthetic speech as natural. The discriminator for calculating the spoofing rates was constructed using natural speech parameters and generated speech parameters of the conventional MGE training. The generation loss and spoofing rates were first calculated with various hyper-parameter $\omega_D$ settings.

Figure 8 shows the results for the generation loss and spoofing rate. As $\omega_D$ increases from 0.0, the generation loss monotonically increases, but from 0.4, we cannot see any tendency. On the other hand, the spoofing rate significantly increases as $\omega_D$ increases from 0.0 to 0.2; from 0.2, the value does not vary much. These results demonstrate that the proposed training algorithm makes the generation loss worse but can train the acoustic models to deceive the discriminator; in other words, although our method does not necessarily decrease the generation error, it tries to reduce the difference between the distributions of natural and generated speech parameters by taking the adversarial loss into account during the training.

2) **Investigation of Convergence in Training:** To investigate the convergence of the proposed training algorithm, we ran the algorithm through 100 iterations. Figure 9 plots the generation loss and adversarial loss for the training and evaluation data. We can see that both loss values are almost monotonically decreased in training. Although the values of evaluation data strongly vary after a few iterations, they can converge after several more iterations.

3) **Subjective Evaluation of Spectral Parameter Generation:** A preference test (AB) test was conducted to evaluate the quality of speech produced by the algorithm. We generated speech samples with three methods:
TABLE IV
ARCHITECTURES OF DNNs USED IN TTS EVALUATIONS. FEED-FORWARD NETWORKS WERE USED FOR ALL ARCHITECTURES. ReLU INDICATES RECTIFIED LINEAR UNIT [32].

|                          | Spectral parameter generation (sections IV-B-1 and IV-B-2) | Spectral and F₀ parameter generation (section IV-B-3) | Duration generation (section IV-B-4) |
|--------------------------|-------------------------------------------------------------|--------------------------------------------------------|-------------------------------------|
| Acoustic models          | 274–3 × 400 (ReLU)–75 (linear)                              | 442–3 × 512 (ReLU)–94 (linear)                         | 442–3 × 512 (ReLU)–94 (linear)      |
| Discriminator            | 25–2 × 200 (ReLU)–1 (sigmoid)                               | 26–3 × 256 (ReLU)–1 (sigmoid)                          | 1–3 × 256 (ReLU)–1 (sigmoid)        |
| Duration models          | N/A                                                         | 439–3 × 256 (ReLU)–1 (linear)                          | 439–3 × 256 (ReLU)–1 (linear)       |

Fig. 8. Parameter generation loss (above) and spoofing rate (below) for various ω₀ for spectral parameter generation in TTS.

MGE: conventional MGE (= Proposed (ω₀ = 0.0))
Proposed (ω₀ = 0.3): spoofing rate > 0.99
Proposed (ω₀ = 1.0): standard setting

Every pair of synthetic speech samples generated by using each method was presented to listeners in random order. Listeners participated in the assessment by using our crowdsourced subjective evaluation systems.

The results are shown in Fig. 10. In Figs. 10(a) and (b), the proposed algorithm outperforms conventional MGE training algorithm in both hyper-parameter settings. Therefore, we can conclude that our algorithm robustly yields significant improvement in terms of speech quality regardless its hyper-parameter setting. Henceforth, we set the hyper-parameter to 1.0 for the following evaluations because Fig. 10(c) shows that the score of “Proposed (ω₀ = 1.0)” was slightly better than that of “Proposed (ω₀ = 0.3).”

4) Subjective Evaluation of F₀ Generation: We evaluated the effect of the proposed algorithm for F₀ generation. We conducted a subjective evaluation using the following three methods:

MGE: conventional MGE
Proposed (sp): proposed algorithm applied only to spectral parameters
Proposed (sp+F0): proposed algorithm applied to spectral and F₀ parameters

Every pair of synthetic speech samples generated by using each method was presented to listeners in random order. Since Fig. 10 has already demonstrated that the proposed algorithm improves synthetic speech quality in terms of generating spectral parameters, we did not compare “Proposed (sp)” with “MGE.” Listeners participated in the assessment by using our crowdsourced subjective evaluation systems.

Figure 11 shows the results. Since the score of “Proposed (sp+F0)” is much higher than those of “Proposed (sp)” and “MGE,” we can confirm the effectiveness of the proposed algorithm for not only spectral parameters but also F₀.
5) **Subjective Evaluation of Duration Generation:** We evaluated the effect of the proposed algorithm for duration generation. We conducted a subjective evaluation using the following three methods:

- **MSE:** conventional MSE
- **Proposed (phoneme):** proposed algorithm applied to phoneme duration
- **Proposed (mora):** proposed algorithm applied to mora duration

The preference AB test was conducted in the same manner as in the previous evaluation described in Section.

The results are shown in Fig. 12. There are no significant differences in the resulting scores. To investigate the reason, we constructed a discriminator that distinguishes conventional MSE and natural speech, and calculated the classification accuracy. We expect that our algorithm works better when the conventional generated parameters are much distinguished from the natural ones. As shown in Fig. 13, the accuracy of the discriminator that uses durations is lower than that of the discriminator that uses spectral parameters and $F_0$. This result infers that distribution compensation by our algorithm does not work well in duration generation. Henceforth, we did not apply the proposed algorithm for generating durations.

6) **Comparison to GV Compensation:** Figure 6 demonstrated that our method compensates the GV of the generated speech parameters. In addition, we investigate whether or not our method improves speech quality more than explicit GV compensation. We applied the post-filtering process [51] to the spectral and $F_0$ parameters generated by the MGE training. A preference AB test with 29 listeners was conducted by using our crowd-sourced subjective evaluation systems.

Figure 14 shows the results. Since the score of “Proposed” is higher than that of the conventional GV post-filter (“MGE-GV”), we can conclude that our method produces more gain in speech quality than the conventional GV compensation.
We adopted the following GANs: W-GAN: Eqs. (18) and (19); JS-GAN: Eqs. (15) and (16); RKL-GAN: Eqs. (12) and (13); KL-GAN: Eqs. (9) and (10); GAN: Eqs. (4) and (5).

As the final investigation regarding TTS, we compared speech qualities of various complicated ones. We conducted a MOS test on speech quality. The synthetic speech generated by using each GAN was presented to listeners in random order. 55 listeners participated in the assessment by using our crowdsourced subjective evaluation systems.

Figure 17 shows the results. We can see that our method works in the case of all divergences except “KL-GAN” and “JS-GAN.” Two points are noteworthy: 1) minimizing KL-divergence (KL-GAN) did not improve synthetic speech quality, but the reversed version (RKL-GAN) worked, and 2) JS-divergence did not work well, but the approximated version (GAN) worked. The best GAN in terms of synthetic speech quality was the W-GAN, whose MOS score was significantly higher than those of the LS-GAN, JS-GAN, and KL-GAN.

C. Experimental Conditions in VC Evaluation

The experimental conditions such as dataset used in the evaluation, speech parameters, pre-processing of data, and training procedure were the same as the previous evaluations except for the dimensionality of spectral parameters and DNN architectures. We constructed DNNs for male-to-male conversion and male-to-female conversion. The hidden layers of the acoustic models and discriminator had $3 \times 512$ units and $3 \times 256$ units, respectively. The 1st-through-59th mel-cepstral coefficients were converted. The input 0th mel-cepstral coefficients were directly used as those of the converted speech. $F_0$ was linearly transformed, and band-aperiodicity was not transformed. Dynamic time warping was used to align total frame lengths of the input and output speech parameters.

We generated speech samples with the conventional MGE training and the proposed training algorithm. We conducted a preference AB test to evaluate the converted speech quality. We presented every pair of converted speech of the two sets in random order and had listeners select the speech sample that sounded better in quality. Similarly, an XAB test on the speaker individuality was conducted using the natural speech as a reference “X.” Eight listeners participated in assessment of male-to-male conversion case, and 27 listeners participated in assessment of male-to-female conversion case using our crowdsourced subjective evaluation systems.
The results of the preference tests on speech quality and speaker individuality are shown in Fig. 18 and Fig. 19 respectively. We can find that our algorithm achieves better scores in speech quality the same as in the TTS evaluations. Moreover, we can see that the proposed algorithm also improves speaker individuality. We expect that the improvements are caused by compensating GVs of the generated speech parameters which affect speaker individuality. These improvements were observed not only in the inter-gender case but also cross-gender case. Therefore, we have also demonstrated the effectiveness of the algorithm in DNN-based VC.

V. CONCLUSION

In this paper, we proposed a novel training algorithm for deep neural network (DNN)-based high-quality statistical parametric speech synthesis. The algorithm incorporates a framework of generative adversarial networks (GANs), which adversarially train generator networks and discriminator networks. In the case of proposed algorithm, acoustic models of speech synthesis are trained to deceive the discriminator that distinguishes natural and synthetic speech. Since the GAN framework minimizes the difference in distributions of natural and generated data, the acoustic models are trained to not only minimize the generation loss but also make the parameter distribution of the generated speech parameters close to that of natural speech. This is a pioneering method of GAN-based speech synthesis and can be applied not only statistical parametric approaches but also the ones such as glottal waveform synthesis. We found that our algorithm compensated not only global variance but also correlation among generated speech parameters. Experimental evaluations were conducted in both DNN-based text-to-speech (TTS) synthesis and voice conversion (VC). The results demonstrate that the proposed algorithm yields significant improvements in terms of speech quality in both TTS and VC regardless of its hyper-parameter settings. We also found that the proposed algorithm incorporating the Wasserstein GAN improved synthetic speech quality the most in comparison with various GANs. In future work, we will further investigate the behavior in relation to the hyper-parameter settings, adopt feature functions which are more effective to detect synthetic speech than the identity function, and devise discriminator models with linguistic dependencies.

REFERENCES

[1] H. Zen, K. Tokuda, and A. Black, “Statistical parametric speech synthesis,” Speech Communication, vol. 51, no. 11, pp. 1039–1064, 2009.
[2] Y. Sagisaka, “Speech synthesis by rule using an optimal selection of non-uniform synthesis units,” in Proc. ICASSP, New York, U.S.A., Apr. 1988, pp. 679–682.
[3] Y. Stylianou, O. Cappé, and E. Moulines, “Continuous probabilistic transform for voice conversion,” IEEE Transactions on Speech and Audio Processing, vol. 6, no. 2, pp. 131–142, Mar. 1998.
[4] Z.-H. Ling, S. Y. Kang, H. Zen, A. Senior, M. Schuster, X. J. Qian, H. Meng, and L. Deng, “Deep learning for acoustic modeling in parametric speech generation: A systematic review of existing techniques and future trends,” IEEE Signal Processing Magazine, vol. 32, no. 3, pp. 35–52, May 2015.
[5] K. Tokuda, Y. Nankaku, T. Toda, H. Zen, J. Yamagishi, and K. Oura, “Speech synthesis based on hidden Markov models,” Proceedings of the IEEE, vol. 101, no. 5, pp. 1234–1252, Apr. 2013.
[6] T. Toda, A. W. Black, and K. Tokuda, “Voice conversion based on maximum likelihood estimation of spectral parameter trajectory,” IEEE Transactions on Audio, Speech, and Language Processing, vol. 15, no. 8, pp. 2222–2235, Nov. 2007.
[7] Y. J. Wu and R. H. Wang, “Minimum generation error training for HMM-based speech synthesis,” in Proc. ICASSP, Toulouse, France, May 2006, pp. 89–92.
[8] Z. Wu and S. King, “Improving trajectory modeling for DNN-based speech synthesis by using stacked bottleneck features and minimum trajectory error training,” IEEE Transactions on Audio, Speech, and Language Processing, vol. 24, no. 7, pp. 1255–1265, Jul. 2016.
[9] T. Toda, L. H. Chen, D. Saito, F. Villavicencio, M. Wester, Z. Wu, and J. Yamagishi, “The Voice Conversion Challenge 2016,” in Proc. INTERSPEECH, California, U.S.A., Sep. 2016, pp. 1632–1636.
[10] Y. Ijima, T. Asami, and H. Mizuno, “Objective evaluation using association between dimensions within spectral features for statistical parametric speech synthesis,” in Proc. INTERSPEECH, California, U.S.A., Sep. 2016, pp. 337–341.
[11] Y. Ohtani, M. Tamura, M. Morita, T. Kagoshima, and M. Akamine, “Histogram-based spectral equalization for HMM-based speech synthesis using mel-LSF,” in Proc. INTERSPEECH, Portland, U.S.A., Sep. 2012.
[12] S. Takamichi, T. Toda, A. W. Black, G. Neubig, S. Sakti, and S. Nakamura, “Postfilters to modify the modulation spectrum for statistical parametric speech synthesis,” IEEE Transactions on Audio, Speech, and Language Processing, vol. 24, no. 4, pp. 755–767, Apr. 2016.
[13] S. Takamichi, T. Toda, A. W. Black, and S. Nakamura, “Modulation spectrum-constrained trajectory training algorithm for GMM-based voice conversion,” in Proc. ICASSP, Brisbane, Australia, Apr. 2015, pp. 4859–4863.
Shinnosuke Takamichi received his B.E. from Nagaoa University of Technology, Japan, in 2011 and his M.E. and Ph.D. from the Graduate School of Information Science, Nara Institute of Science and Technology (NAIST), Japan, in 2013 and 2016, respectively. He was a short-term researcher at NICT, Kyoto, Japan in 2013, a visiting researcher of Carnegie Mellon University (CMU) in the United States from 2014 to 2015, and a Research Fellow (DC2) of Japan Society for the Promotion of Science, from 2014 to 2016. He is currently a Project Research Associate of the University of Tokyo. He has received more than ten paper/achievement awards including the 8th Outstanding Student Paper Award from IEEE Japan Chapter SPS and the Itakura Prize Innovative Young Researcher Award. He is a member of ASJ, IPSJ, ISCA and IEEE SPS.

Hiroshi Saruwatari received B.E., M.E., and Ph.D. degrees from Nagoya University, Nagoya, Japan in 1991, 1993, and 2000, respectively. He joined the Intelligent System Laboratory, SECOM Co., Ltd., Tokyo, Japan in 1993, where he was engaged in research on the ultrasonic array system for acoustic imaging. He is currently a Professor at Graduate School of Information Science and Technology, the University of Tokyo, Tokyo. His research interests include noise reduction, array signal processing, blind source separation, and sound field reproduction. He received paper awards from the IEICE in 2001 and 2006, from the Telecommunications Advancement Foundation in 2004 and 2009, and at the IEEE-IROS2005 in 2006. He won the first prize at the IEEE MLSP2007 Data Analysis Competition for BSS. He is a Member of the IEICE, Japan VR Society, and the ASJ.