Department of Creative Informatics
Graduate School of Information Science and Technology
THE UNIVERSITY OF TOKYO

Master's Thesis

# High-Quality Statistical Parametric Speech Synthesis Using Generative Adversarial Networks

敵対的学習を用いた高品質な統計的パラメトリック音声合成

**Yuki Saito**
齋藤 佑樹

Supervisor: Professor Hiroshi Saruwatari

January 2018

# Abstract

Speech synthesis is a technique for artificially synthesizing natural human speech. Text-to-speech (TTS) is a technique for synthesizing speech from text, and voice conversion (VC) is a technique for synthesizing speech from another one while preserving the linguistic information of the original speech. With these speech synthesis techniques, a method that not only synthesizes natural-sounding speech but also easily controls the characteristics of the synthetic speech is required. Statistical parametric speech synthesis (SPSS), which is covered in this thesis, is a method for constructing acoustic models representing the relationship between the input features (i.e., linguistic features of text in TTS and source speech parameters in VC) and the speech parameters. Although this method has the flexibility needed to control the characteristics of synthetic speech, the quality of the synthetic speech is low compared with that of natural speech. This is primarily due to the over-smoothing effect often observed in generated speech parameters. This effect can be alleviated by compensating for the differences between the natural and synthetic speech such as the global variance (i.e., the second moment of a distribution) and the modulation spectrum of the speech parameter sequences. However, quality degradation is still a critical problem. For further improvement of speech quality in SPSS, this thesis presents a novel algorithm for training acoustic models for SPSS using generative adversarial networks (GANs). GANs consist of two deep neural networks (DNNs): one works as a discriminator to distinguish natural and generated samples and the other works as a generator to deceive the discriminator. This thesis defines a new training criterion for acoustic models based on this framework. The criterion is the weighted sum of the conventional minimum generation error loss of the speech parameters and the adversarial loss to make 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 samples, the proposed algorithm effectively alleviates the effect of over-smoothing. The proposed algorithm can be regarded as a generalization of the conventional method using explicit modeling of analytically derived features such as the global variance and modulation spectrum because it effectively minimizes the divergence without explicit statistical modeling. The discriminator used in the proposed algorithm can be interpreted as anti-spoofing, i.e., as a technique for detecting synthetic speech and preventing voice spoofing attacks. Accordingly, techniques and ideas concerning anti-spoofing can be applied to the proposed training algorithm. This thesis investigates the effectiveness of the proposed algorithm in two domains: 1) DNN-based SPSS using vocoder-derived speech parameters and 2) that using short-term Fourier transform spectra, which is becoming one of the mainstreams of SPSS research. Experimental results demonstrate that the proposed algorithm improves synthetic speech quality.

# 概要

音声合成とは，コンピュータを用いて音声を人工的に生成する技術である．特に，テキストから音声を生成する技術をテキスト音声合成といい，入力された音声の言語情報を保持しつつ，非言語情報を変換する技術を音声変換という．これらの音声合成技術には，高品質な合成音声を生成でき，かつ，生成される合成音声の声質を容易に制御できる手法が求められる．本論文で対象とする統計的パラメトリック音声合成は，入力特徴量と音声パラメータの統計的な対応付けを表現する音響モデルを学習させる手法である．この手法では，合成音声の声質制御が容易だが，統計処理に起因する合成音声パラメータの過剰な平滑化により，合成音声の音質が人間の自然音声と比較して著しく劣化するという問題がある．過剰な平滑化を定量化する指標として，これまでに，合成音声パラメータの系列内変動（分布の 2 次モーメント）や，変調スペクトルなどの解析的特徴量が提案されており，これらを補償することによる合成音声の音質改善が確認されているが，合成音声の音質は未だに低い．本論文では，統計的パラメトリック音声合成のさらなる音質改善を目的として，画像生成の分野において有効な手法として知られている敵対的学習の枠組みを用いた音響モデル学習法を新たに提案する．敵対的学習は，識別モデルと生成モデルの 2 つの deep neural networks を学習させる手法である．識別モデルは，真のデータと生成モデルにより生成されたデータを識別するように学習される．一方で，生成モデルは，識別モデルを詐称するデータを生成するように学習される．提案手法における音響モデル学習時の学習基準は，従来の音声パラメータの生成誤差と，敵対的学習に由来する，識別モデルを詐称する損失の重み付き和として表現される．提案手法では，自然音声パラメータと合成音声パラメータの分布間距離最小化を考慮して音響モデルを学習させるため，過剰な平滑化を緩和できる．これは，従来の解析的特徴量を明示的に補償する手法の拡張と解釈でき，合成音声のさらなる音質改善が期待できる．また，提案手法において導入される識別モデルは，合成音声による声のなりすましを防ぐ anti-spoofing として解釈できるため，anti-spoofing の知見を取り入れた音声合成も実現できる．本論文では，提案手法の有効性を (1) ボコーダパラメータを用いた統計的パラメトリック音声合成，及び (2) 近年の主流となることが予想される音声合成方式の 1 つである，短時間フーリエ変換スペクトルを用いた統計的パラメトリック音声合成において調査し，実験的評価によりその有効性を示す．

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Background

Although people can communicate with each other in several ways, the most natural way is speaking. The ideas in their minds are conveyed by speech waveforms, which include not only linguistic information but also para-/non-linguistic information such as the speaker's emotions and attitude. Speech synthesis is a technology for mimicking speech behavior by using computers and therefore extending speech communication.

This thesis covers two techniques of speech synthesis; text-to-speech (TTS) [2] and voice conversion (VC) [3], which are illustrated in Fig. 1.1. TTS is a technique for synthesizing speech from a given text, which is typically applied to speech interface for computers and smartphones (i.e., man-machine interface). VC is a technique for synthesizing speech from another one while preserving the linguistic information of the original speech, which can be used to overcome the limitation in the human speech production systems. These techniques can not only assist natural speech communication (e.g., speech translation systems to remove language differences [4, 5]) but also offer us some entertainments such as singing voice conversion systems [6, 7]. Figure 1.2 shows some applications of the speech synthesis techniques. With these techniques, a method that not only synthesizes natural-sounding speech but also enables the characteristics of synthetic speech to be easily controlled is required.

Statistical parametric speech synthesis (SPSS) [8], the topic of this thesis, is a method for learning a statistical mapping from input (i.e., text in TTS and source speech in VC) to output speech. In SPSS, several steps are taken to synthesize the desired speech. First, the input features (i.e., the linguistic features of the text in TTS and source speech parameters in VC) and the output speech parameters are extracted from a training dataset. Then, acoustic models representing the relationship between the input features and output speech parameters are constructed. Deep neural networks (DNNs) have recently come to be used as acoustic models [9] because they can model the relationships between the input features and speech parameters more accurately than conventional hidden Markov models (HMMs) [10] and Gaussian mixture models (GMMs) [11]. Finally, a synthetic

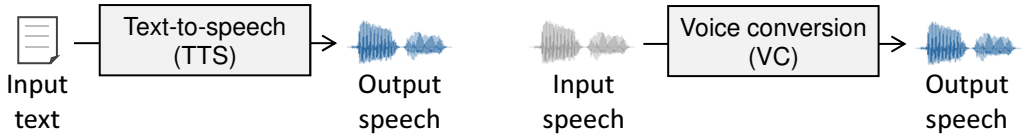


Fig. 1.1: Two speech synthesis techniques covered in this thesis. The difference between TTS and VC is the information input to the speech synthesis systems.

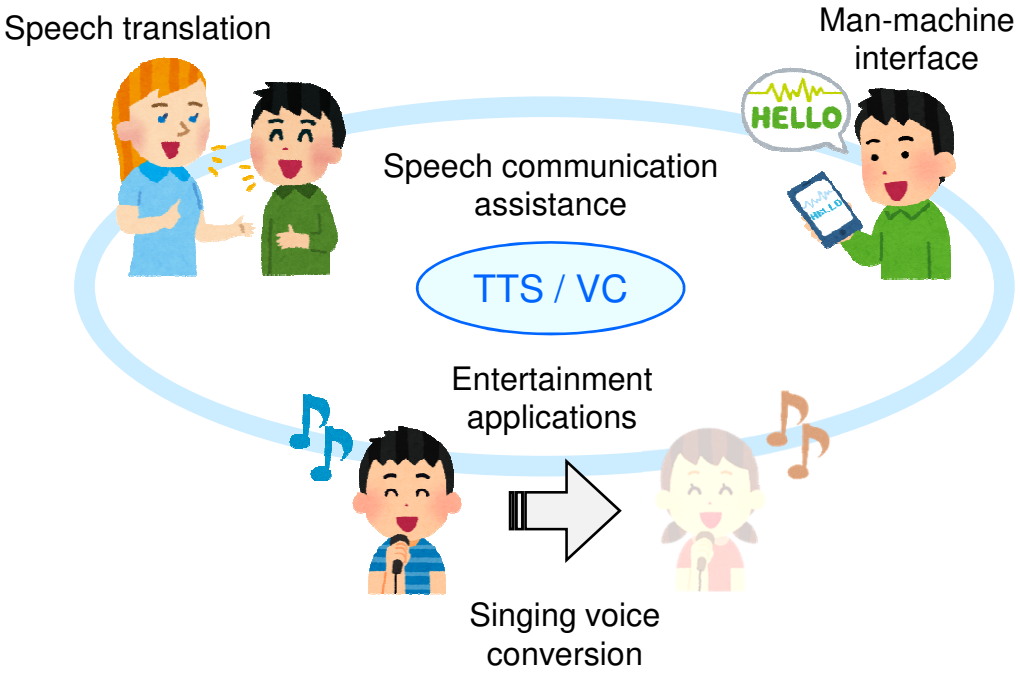


Fig. 1.2: Applications of speech synthesis techniques.

speech waveform is synthesized from speech parameters predicted by the acoustic models. The acoustic modeling techniques used in SPSS for generating high-quality speech parameters have been widely studied since they can be used for both TTS and VC. Although SPSS supports flexible control characteristics of the synthetic speech, speech quality is degraded.

A primary cause of the quality degradation is over-smoothing [8, 12] of the generated speech parameters, which removes fine structures of the natural speech parameters and makes the synthetic speech sound muffled. One way to alleviate this effect is to reduce the differences between the natural and generated speech parameters. This can be done by, for example, modeling the probability distributions in a parametric [11] or non-parametric [13] way in the acoustic model training and then generating or transforming the synthetic speech parameters by using the distributions. A effective approach is to use analytically derived features correlated to the quality degradation. Global variance (GV) [11] and modulation spectrum (MS) [14] are well-known derived features for reproducing natural statistics. These features work as a constraint in the training stage [15, 16]. Nose and Ito [17] and Takamichi et al. [15] proposed methods for reducing the differences between

natural and generated GV and MS Gaussian distributions. However, quality degradation is still a critical problem.

Not only the over-smoothing effect but also speech parameter extraction and speech waveform synthesis affect synthetic speech quality. High-quality vocoder systems have played an important role in both parameter extraction and waveform synthesis. In conventional SPSS using vocoder systems, vocoder-derived speech parameters representing the characteristics of a vocal cord and vocal tract are extracted from a speech waveform. The characteristics of synthetic speech can be easily controlled by using the derived speech parameters. However, the quality degradation due to vocoder-based parameterization in state-of-the-art DNN-based speech synthesis has also become a critical problem. For example, a vocoder process to synthesize a speech waveform causes buzziness in synthetic speech and degrades quality considerably. The most straighforward approach to avoid the vocoder-based parameterization is to generate low-level features such as short-term Fourier transform (STFT) spectra [18] and speech waveforms [19, 20] before vocoder-based parameterization. Such vocoder-free SPSS can achieve higher quality in the synthetic speech than the conventional SPSS using vocoders. However, in addition to the over-smoothing effect, difficulty in acoustic model training due to the high dimensionality of the spectral amplitudes causes a significant quality degradation problem.

## 1.2 Thesis Scope

This thesis presents novel algorithms for overcoming the quality degradation caused by the over-smoothing of the generated speech parameters. They use generative adversarial networks (GANs) to train acoustic models for SPSS. GANs consist of two DNNs: a discriminator to distinguish natural and generated samples and a generator to deceive the discriminator. A new training criterion is defined for acoustic models that is based on this framework; the criterion is the weighted sum of the conventional minimum generation error (MGE) loss of the speech parameters and the adversarial loss, which makes the discriminator recognize the generated speech parameters as natural. Since the objective of GANs is to minimize the divergence (i.e., the distribution difference) between natural and generated speech parameters, the proposed algorithm effectively alleviates the effect of the over-smoothing. It 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. The discriminator used in the proposed algorithm can be interpreted as anti-spoofing, i.e., as a technique for detecting synthetic speech and preventing voice spoofing attacks. Accordingly, techniques and ideas concerning anti-spoofing can be applied to the training algorithm. This thesis first evaluates the effectiveness of the proposed algorithm in DNN-based SPSS using vocoder-derived speech parameters. The evaluations also investigate the effect of the divergence of various types of GANs, including image-processing-related and speech-processing-related GANs.

This thesis also extends the proposed algorithm to DNN-based SPSS using STFT spectra. To overcome the difficulty in training acoustic models because of the complex distribution of STFT spectral amplitudes, a training algorithm using low-resolution GANs is proposed. Through a pooling layer along with a frequency axis, spectral amplitudes are converted into low-resolution spectra. The training criterion for the acoustic models is the weighted sum of the mean squared error (MSE) between the natural and generated spectral amplitudes in the original frequency resolution and the adversarial loss using a discriminator for the low-frequency-resolution GANs. GANs with low resolution can be regarded as one compensating for the difference between the spectral envelopes of the natural and synthetic speech because the low-resolution spectra approximately emulate filter banks. Since the spectral envelopes are dominant features in the quality of synthetic speech and the evaluation using the vocoder-derived speech parameters revealed that the GANs are particularly effective for generating spectral parameters, using GANs with low resolution should improve the speech quality better than using GANs with the original resolution. The algorithm using low-resolution GANs can be extended to one using low-resolution GANs and original-resolution GANs, which is expected to compensate for not only the differences in spectral envelopes but also fine structures (i.e., the excitation parameters) of the natural and generated STFT spectral amplitudes.

## 1.3 Thesis Overview

This thesis is organized as follows. Chapter 2 briefly reviews the framework of conventional DNN-based SPSS. Chapter 3 introduces GANs and presents an algorithm for acoustic model training of SPSS using GANs. Experimental results for the proposed algorithm using vocoder-derived speech parameters are also presented. Chapter 4 extends the algorithm to vocoder-free SPSS using STFT spectra and demonstrates its effectiveness. Chapter 5 summarizes the key points and mentions future work. Figure 1.3 depicts the outline of the proposed methods in the SPSS techniques.

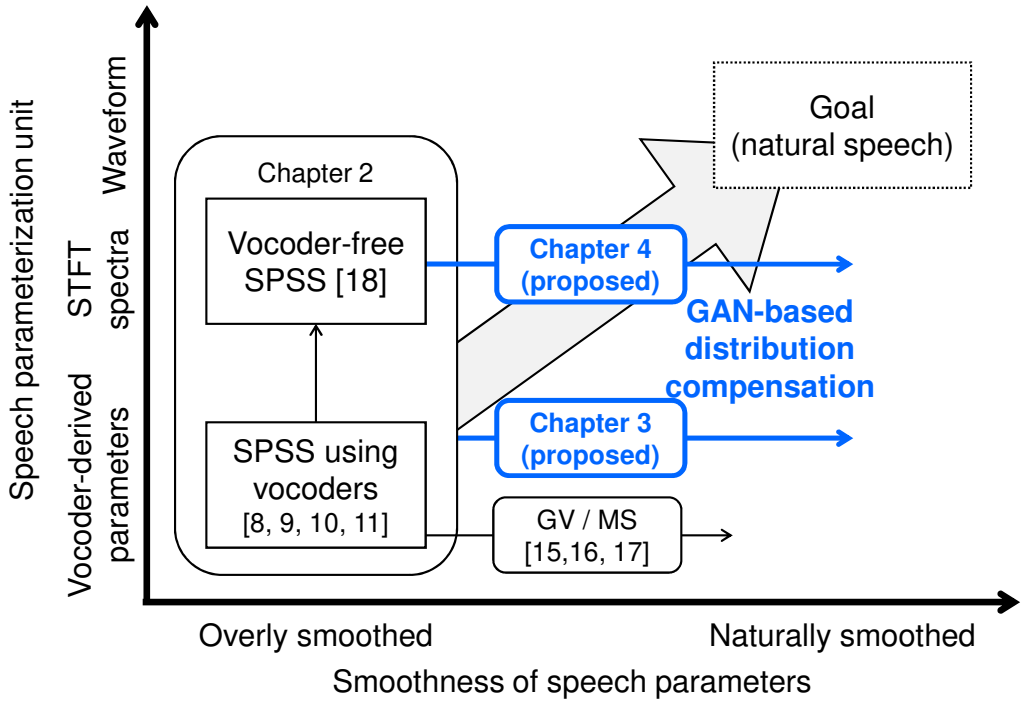


Fig. 1.3: Relation between conventional and proposed methods for SPSS.

# Chapter 2

# Statistical Parametric Speech Synthesis Using Deep Neural Networks

## 2.1 Introduction

SPSS consists of several steps for synthesizing a speech waveform from input information. Figure 2.1 shows the basic framework of SPSS using DNNs. First, the input features (i.e., linguistic information of the given text in TTS and speech parameters of the source speech in VC) and output speech parameters are extracted in the feature analysis. Next, acoustic models representing the relationships between the input features and output speech parameters are trained. In this thesis, DNNs are used as the acoustic models. Then, speech parameters are generated from the input features by using the trained acoustic models. Finally, a speech waveform is synthesized using the generated speech parameters.

This chapter is organized as follows. Section 2.2 describes SPSS processing using vocoder systems. Section 2.3 presents techniques for vocoder-free SPSS using STFT spectra. Section 2.4 summarizes this chapter.

## 2.2 DNN-based Statistical Parametric Speech Synthesis Using Vocoders

### 2.2.1 Feature Analysis

#### Speech Analysis

To facilitate control of the characteristics of the synthetic speech, parameters representing the vocal tract and vocal cord features are extracted from the speech waveform. On the basis of a source-filter model, a speech signal is represented as convolutions of two components: spectral parameters representing the vocal tract features and excitation parameters representing the vocal cord features. These parameters are extracted from

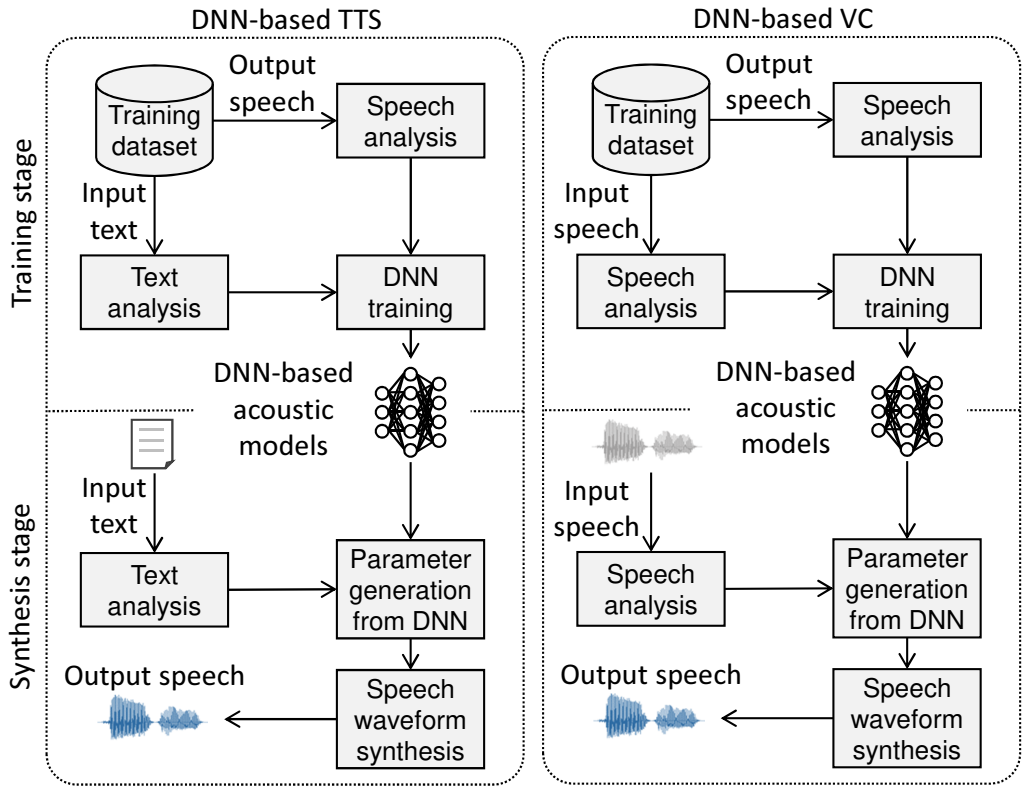


Fig. 2.1: Flowcharts for SPSS using DNNs. DNN-based acoustic models are constructed to represent the input features and output speech parameters.

the STFT power spectra of the speech signal. The excitation parameters are further decomposed into periodic factors typically represented as the fundamental frequency ($F_0$) and aperiodic factors [21]. Figure 2.2 shows an example of STFT spectra and its spectral envelope of a speech signal.

As the dimensionality of the spectral parameters tends to be high, a dimensionality reduction technique is applied before acoustic modeling. A commonly used technique uses mel-cepstral coefficients [22], which take the perceptual effects of human listening in lower frequency components into account for the dimensionality reduction.

In modeling $F_0$, the difference in value between the voiced regions (V) and unvoiced regions (U) must be considered. Continuous $F_0$ modeling [23] was proposed to efficiently represent the $F_0$ parameters. It uses one-dimensional continuous values to represent the observed log $F_0$ and one-dimensional discrete values to represent U/V (0 for U and 1 for V). The values of log $F_0$ observed in unvoiced regions are estimated using SPLINE interpolation. Figure 2.3 shows an example of a continuous $F_0$ sequence and U/V labels.

In this thesis, the STRAIGHT vocoder [24] is used to extract the speech parameters. Although such vocoders have high quality, deployment of speech synthesis systems using the STRAIGHT vocoder is limited due to its patent protection. Recently, the freely available WORLD vocoder [25, 26] was proposed and gets used widely instead of the STRAIGHT vocoder.

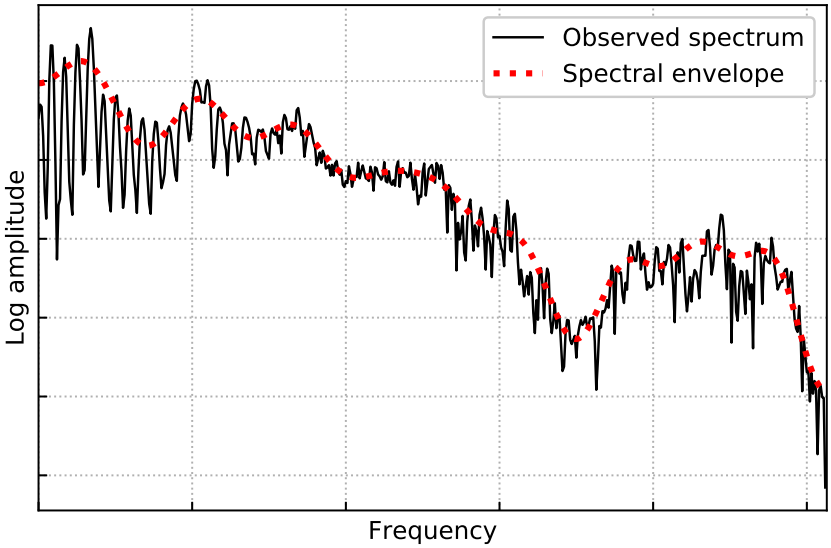


Fig. 2.2: Example STFT spectrum of speech signal represented as the product of the spectral envelope and the excitation parameters in the frequency domain.

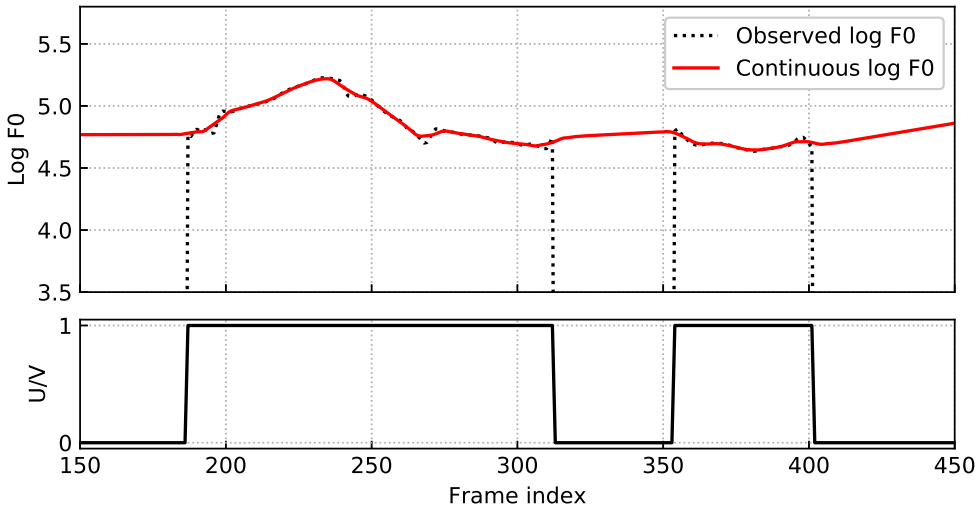


Fig. 2.3: Example of continuous $F_0$ modeling. The continuous $F_0$ sequence and U/V labels are separately modeled.

### Text Analysis

In TTS, linguistic features are extracted from the input text. This thesis focuses on Japanese TTS systems [27]. Japanese linguistic features consist of phoneme, accent type, word, part-of-speech, breath group, and sentence length. A text analyzer such as MeCab [28] is used to extract these features. They are represented as multi-dimensional

vectors including categorical factors (e.g., phoneme identity and accent type) and numeric factors (e.g., the total number of phonemes and sentence length). Because Japanese is a mora-timed language (i.e., mora isochrony), the mora features as well as the phoneme features are included in the linguistic features.

#### Temporal Alignment

The lengths of the input features and output speech parameters are aligned for the acoustic modeling. In TTS, the lengths of the linguistic features are much shorter than those of the speech parameters. Thus, each linguistic feature is duplicated to align its length with the corresponding speech parameters. The Viterbi algorithm using hidden Markov models is used to obtain the phoneme durations. Figure 2.4 shows an example of feature alignment. In VC, the lengths of the source and target speech parameters are aligned using the dynamic time warping (DTW) algorithm [11]. Figure 2.5 shows a conceptual diagram of the DTW algorithm.

### 2.2.2 Acoustic Model Training

#### General Purpose

Acoustic models parameterized by $\theta_{\mathrm{G}}$ define the mapping $\boldsymbol{y} = \boldsymbol{G}(\boldsymbol{x}; \theta_{\mathrm{G}})$ from input features $\boldsymbol{x}$ to output speech parameters $\boldsymbol{y}$. $\boldsymbol{x} = [\boldsymbol{x}_1^\top, \cdots, \boldsymbol{x}_t^\top, \cdots, \boldsymbol{x}_T^\top]^\top$ is an input feature sequence and $\boldsymbol{y} = [\boldsymbol{y}_1^\top, \cdots, \boldsymbol{y}_t^\top, \cdots, \boldsymbol{y}_T^\top]^\top$ is an output speech parameter sequence, where $t$ and $T$ denote the frame index and total frame length, respectively. $\boldsymbol{x}_t = [x_t(1), \cdots, x_t(D_x)]^\top$ and $\boldsymbol{y}_t = [y_t(1), \cdots, y_t(D_y)]^\top$ are a $D_x$-dimensional input feature vector and a $D_y$-dimensional output speech parameter vector at frame $t$, respectively. The goal of acoustic model training is to estimate model parameters $\theta_{\mathrm{G}}$ by using training dataset $\{(\boldsymbol{x}_n, \boldsymbol{y}_n)\}_{n=1}^N$, which includes $N$ pairs of input features and output speech parameter.

#### Static-dynamic Feature Modeling

To take into account temporal continuity, the static-dynamic features of the speech parameters are modeled by acoustic models. Let $\boldsymbol{Y}_t = [\boldsymbol{y}_t^\top, \Delta\boldsymbol{y}_t^\top, \Delta\Delta\boldsymbol{y}_t^\top]^\top$ be a static-dynamic feature vector at frame $t$; $\Delta\boldsymbol{y}_t$ and $\Delta\Delta\boldsymbol{y}_t$ are dynamic features calculated using

$$\Delta\boldsymbol{y}_t = \frac{1}{2}\boldsymbol{y}_{t+1} - \frac{1}{2}\boldsymbol{y}_{t-1}, \tag{2.1}$$

$$\Delta\Delta\boldsymbol{y}_t = \boldsymbol{y}_{t+1} - 2\boldsymbol{y}_t + \boldsymbol{y}_{t-1}. \tag{2.2}$$

A static-dynamic feature sequence $\boldsymbol{Y} = [\boldsymbol{Y}_1^\top, \cdots, \boldsymbol{Y}_t^\top, \cdots, \boldsymbol{Y}_T^\top]^\top$ is calculated as $\boldsymbol{Y} = \boldsymbol{M}\boldsymbol{y}$, where $\boldsymbol{M}$ is a $3D_yT$-by-$D_yT$ matrix used to calculate the dynamic features [10]. Figure 2.6 shows the matrix computation used to obtain the static-dynamic feature sequence.

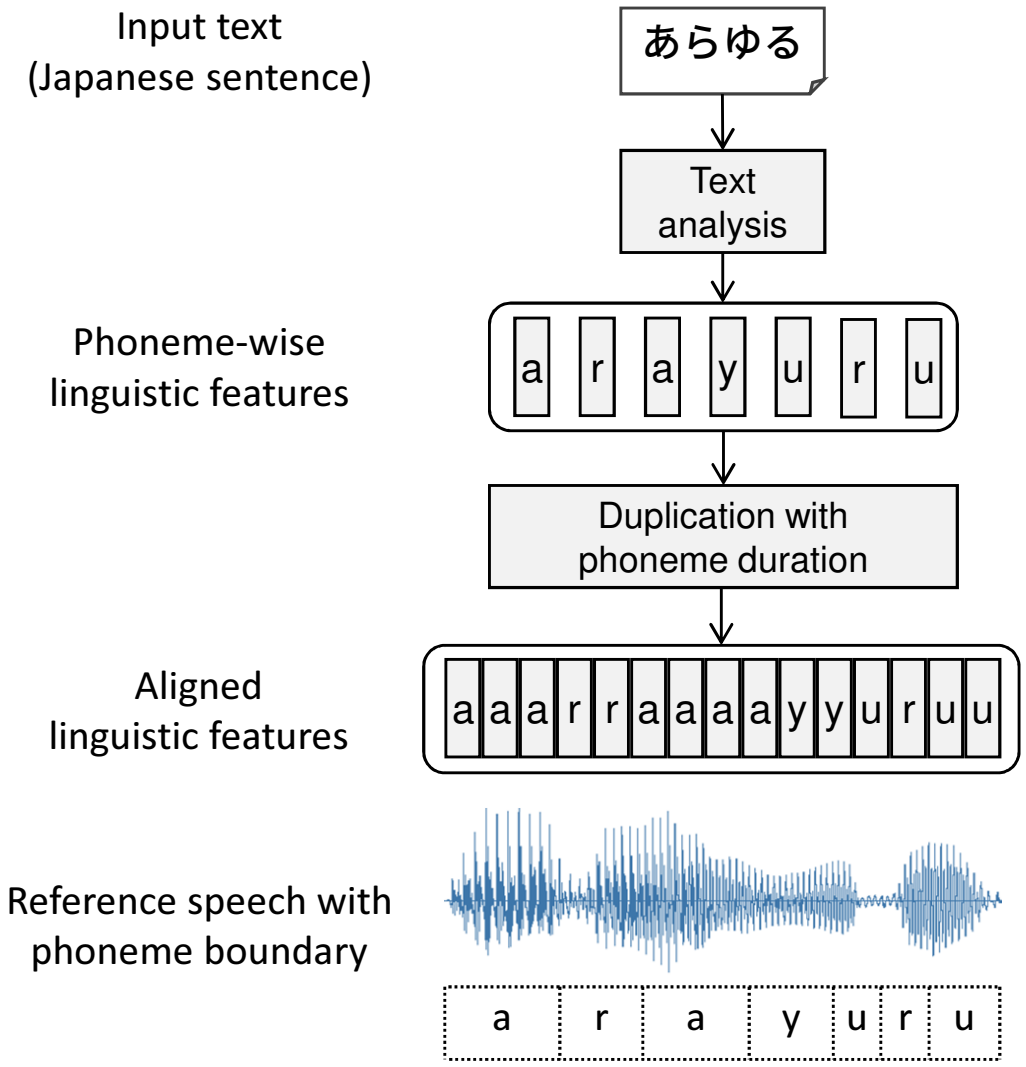


Fig. 2.4: Temporal alignment of features in TTS. Given the phoneme boundary, the phoneme-wise linguistic features are duplicated to align its length with the corresponding speech parameters.

### Acoustic Modeling in TTS

In TTS, besides constructing acoustic models to generate the speech parameters, duration models to predict phoneme durations from linguistic features need to be constructed. After predicting the phoneme durations, the acoustic models for speech parameters predict the joint vector of mel-cepstral coefficients, continuous log $F_0$, U/N, and aperiodic components. Figure 2.7 shows the acoustic models representing the relationships between the linguistic features and speech parameters.

### Acoustic Modeling In VC

In VC, acoustic models predict the static-dynamic features of the mel-cepstral coefficients of the target speech on the basis of those of the source speech. $F_0$ is often linearly converted

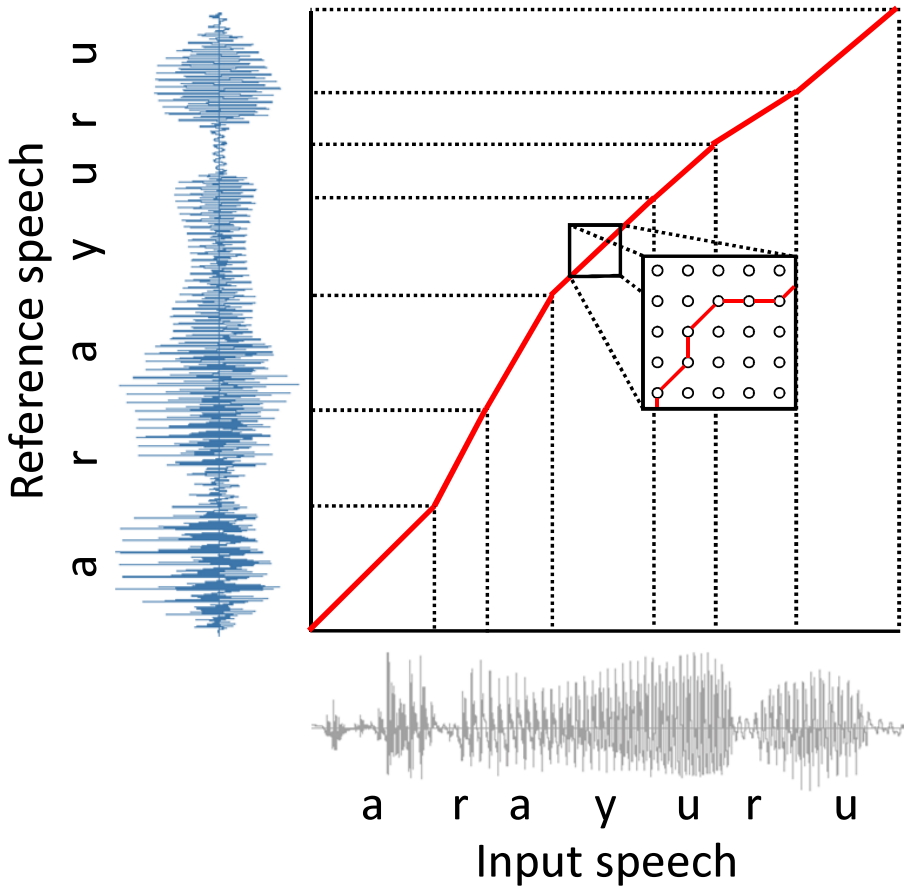


Fig. 2.5: DTW algorithm for feature alignment in VC. The phoneme boundaries of the input and reference speech are superimposed for clear visualization.

by using the statistics of the $F_0$ sequences of the source and target speech.

Since the domains of the input and output features of the acoustic models are the same in VC, the models can be constructed to represent the mapping from the input features to the difference between the two features. In VC using spectral differentials [29], acoustic models are used to represent $\boldsymbol{y} - \boldsymbol{x} = \boldsymbol{G}(\boldsymbol{x}; \theta_{\mathrm{G}})$, rather than $\boldsymbol{y} = \boldsymbol{G}(\boldsymbol{x}; \theta_{\mathrm{G}})$.

### DNN Architectures for Acoustic Models

A DNN is an artificial neural network that has more than one hidden layer between its input layer and output layer [30] that provides a unified framework for acoustic modeling in both TTS and VC. Although there are many architectures, two commonly used architectures are used in this thesis: the Feed-Forward DNNs [31, 32] and the long-short term memory (LSTM) [33, 34].

The Feed-Forward DNN is the foundation of every DNN that transforms an input vector into an output vector through stacked nonlinear transformations. The layer-wise nonlinear transformations of the DNN are defined as element-wise activation functions $g^{(l)}(\cdot), l \in \{1, \cdots, L+1\}$, where $l$ and $L$ denote the layer index and number of layers in

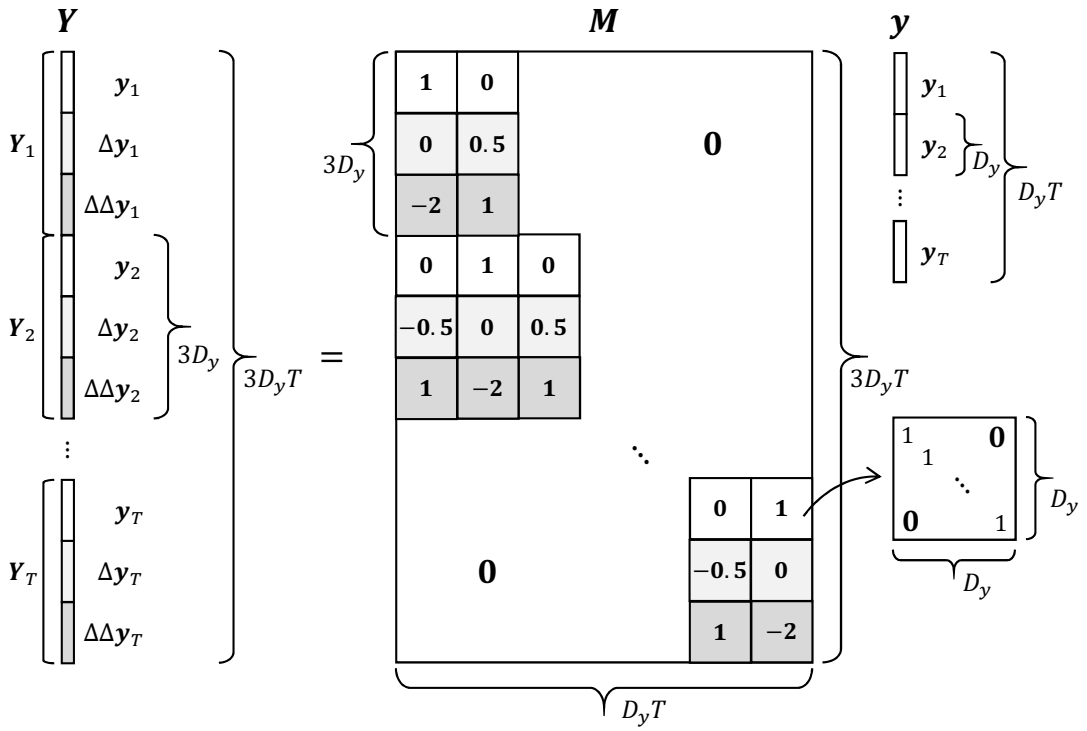


Fig. 2.6: Matrix computation used to obtain static-dynamic feature sequence.

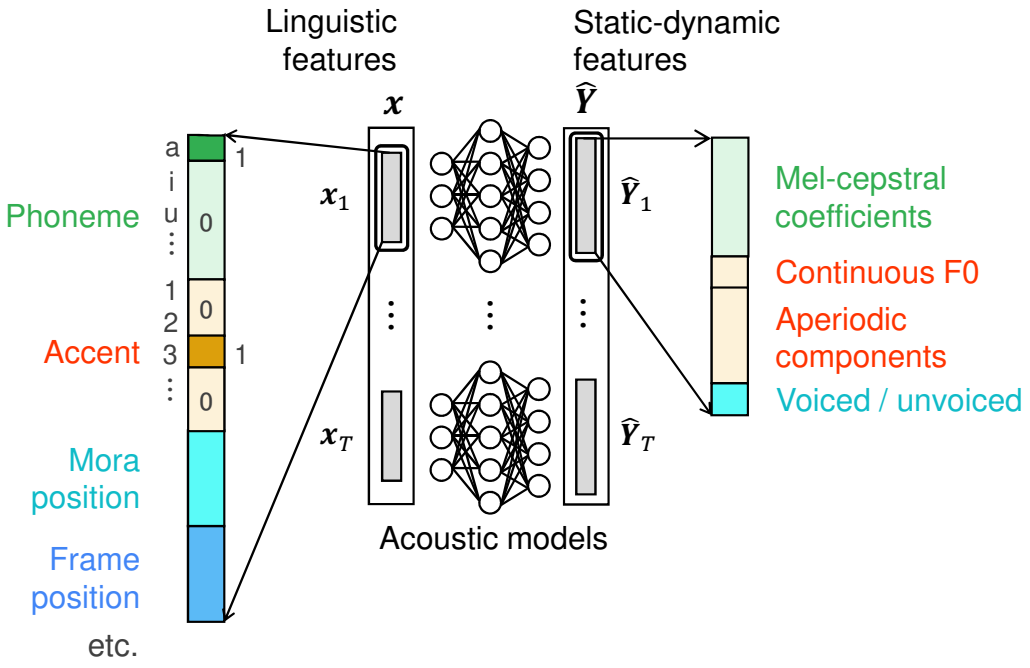


Fig. 2.7: Acoustic models used in TTS. The frame-wise static-dynamic features are predicted from the corresponding linguistic features using the DNN-based acoustic models.

the Feed-Forward DNN, respectively. The definition of the activation function plays an important role in the DNN framework. For the hidden layers ($l = 1, \cdots, L$), the rectified linear unit (ReLU) function [35], defined as $g^{(l)}(\boldsymbol{z}^{(l)}) = \max\{0, \boldsymbol{z}^{(l)}\}$, is often used as

the activation function. For the output layer ($l = L + 1$), the linear function defined as $g^{(L+1)}(\boldsymbol{z}^{(L+1)}) = \boldsymbol{z}^{(L+1)}$ is used as the activation function in regression problems, i.e., TTS and VC. The model parameters of the Feed-Forward DNN, i.e., the weight matrices and bias vectors of each hidden layer, are updated by supervised learning using the backpropagation (BP) algorithm [36]. First, an output vector $\hat{\boldsymbol{y}}$ is predicted from input vector $\boldsymbol{x}$ through the DNN. Then, a defined loss function $L(\boldsymbol{y}, \hat{\boldsymbol{y}})$ is computed with the target vector $\boldsymbol{y}$ and the predicted vector $\hat{\boldsymbol{y}}$. Finally, the model parameters are updated using the stochastic gradient descent (SGD) algorithm with the gradient $\boldsymbol{\nabla}_{\theta_{\mathrm{G}}} L(\boldsymbol{y}, \hat{\boldsymbol{y}})$.

The LSTM is one of the most popular recurrent neural network architectures that can learn sequence modeling. It has a block consisting of a memory cell, input gate, output gate, and forget gate to learn long-short-term dependencies. The memory cell stores information about the previous sequence, and the forget gate controls the weights for old values stored in the memory cell. The input and output gates control the weights for new values of the memory cell and the output values of the LSTM, respectively. The model parameters of the LSTM, i.e., weight matrices and bias vectors of the memory cell and the gates, are updated by supervised learning using the backpropagation through time (BPTT) algorithm [37]. First, the inner loops of the LSTM are unfolded and then the algorithm is run to estimate the gradients of the model parameters.

### Loss Functions for DNN Training

The standard loss function for training DNN-based acoustic models is the mean squared error (MSE) between the natural and generated speech parameters. In DNN-based TTS [31], the acoustic models predict the static-dynamic feature sequence $\hat{\boldsymbol{Y}}$. The loss function for training the models is defined as the MSE between $\boldsymbol{Y}$ and $\hat{\boldsymbol{Y}}$:

$$L_{\mathrm{MSE}}\left(\boldsymbol{Y}, \hat{\boldsymbol{Y}}\right) = \frac{1}{T}\left(\hat{\boldsymbol{Y}} - \boldsymbol{Y}\right)^{\top}\left(\hat{\boldsymbol{Y}} - \boldsymbol{Y}\right). \tag{2.3}$$

The MSE is also used for training the duration models. Let $\boldsymbol{d} = [d_1, \cdots, d_p, \cdots, d_P]^{\top}$ be a natural phoneme duration sequence, and $\hat{\boldsymbol{d}} = [\hat{d}_1, \cdots, \hat{d}_p, \cdots, \hat{d}_P]^{\top}$ be a generated duration sequence, where $p$ is the phoneme index and $P$ is the total number of phonemes. The duration model parameters are updated to minimize $L_{\mathrm{MSE}}(\boldsymbol{d}, \hat{\boldsymbol{d}})$.

To take the static-dynamic constraint of the speech parameters into account, the minimum generation error (MGE) training was proposed [38, 39]. In MGE training, the loss function is defined as the MSE between the natural and generated speech parameters after maximum likelihood parameter generation (MLPG) [1], defined as

$$\begin{aligned} L_{\mathrm{MGE}}\left(\boldsymbol{y}, \hat{\boldsymbol{y}}\right) &= \frac{1}{T}\left(\hat{\boldsymbol{y}} - \boldsymbol{y}\right)^{\top}\left(\hat{\boldsymbol{y}} - \boldsymbol{y}\right) \\ &= \frac{1}{T}\left(\boldsymbol{R}\hat{\boldsymbol{Y}} - \boldsymbol{y}\right)^{\top}\left(\boldsymbol{R}\hat{\boldsymbol{Y}} - \boldsymbol{y}\right), \end{aligned} \tag{2.4}$$

where $\boldsymbol{R}$ is a $D_y T$-by-$3D_y T$ matrix defined as

$$\boldsymbol{R} = \left(\boldsymbol{M}^{\top}\boldsymbol{\Sigma}^{-1}\boldsymbol{M}\right)^{-1}\boldsymbol{M}^{\top}\boldsymbol{\Sigma}^{-1}. \tag{2.5}$$

$\boldsymbol{\Sigma} = \mathrm{diag}[\boldsymbol{\Sigma}_1, \cdots, \boldsymbol{\Sigma}_t, \cdots, \boldsymbol{\Sigma}_T]$ is a $3D_yT$-by-$3D_yT$ covariance matrix, where $\boldsymbol{\Sigma}_t$ is a $3D_y$-by-$3D_y$ covariance matrix at frame $t$. $\boldsymbol{\Sigma}$ is separately estimated using a training dataset. As described by Wu et al. [39], gradient $\nabla_{\hat{Y}} L_{\mathrm{MGE}}(\boldsymbol{y}, \hat{\boldsymbol{y}})$ is given as $\boldsymbol{R}^\top(\hat{\boldsymbol{y}} - \boldsymbol{y})/T$.

### 2.2.3 Speech Parameter Generation

Speech parameters can be generated from the trained acoustic models. In TTS, the phoneme durations of the given linguistic features are first predicted using trained duration models. The static-dynamic feature sequence of the speech parameters is then predicted using trained acoustic models. Finally, MLPG is used to obtain the static features of the speech parameters. In VC, the static-dynamic feature sequence of the target speech parameters or that of the spectral differentials is predicted using the source speech parameters, and MLPG is again used to obtain the static features of the speech parameters.

#### GV Compensation

Although the generated speech parameters after MLPG are temporally smoothed, the fine structures of the natural speech parameters tend to vanish due to over-smoothing, which considerably degrades synthetic speech quality. One way to prevent the fine structures from vanishing is reproducing the statistics of the natural speech. Global variance (GV) compensation [40, 41] is a commonly used technique for improving synthetic speech quality. The GV is defined as the second moment of the speech parameter sequence. A $D_y$-dimensional GV vector of $\boldsymbol{y}$ is calculated using

$$\boldsymbol{v}(\boldsymbol{y}) = [v(1), \cdots, v(d), \cdots, v(D_y)]^\top, \tag{2.6}$$

$$v(d) = \frac{1}{T}\sum_{t=1}^{T}\left(y_t(d) - \langle y(d)\rangle\right)^2, \tag{2.7}$$

$$\langle y(d)\rangle = \frac{1}{T}\sum_{t=1}^{T} y_t(d). \tag{2.8}$$

The GV of the generated speech parameter sequence tends to be smaller than that of the natural speech parameter sequence. The synthetic speech quality can be improved by compensating for the difference between natural and generated GVs. The generated speech parameters after the GV compensation are calculated using

$$\hat{y}_t^{(\mathrm{GV})}(d) = \sqrt{\frac{\mu^{(\mathrm{GV})}(d)}{\hat{\mu}^{(\mathrm{GV})}(d)}}\left\{\hat{y}_t(d) - \langle \hat{y}(d)\rangle\right\} + \langle \hat{y}(d)\rangle, \tag{2.9}$$

where $\mu^{(\mathrm{GV})}(d)$ and $\hat{\mu}^{(\mathrm{GV})}(d)$ are the $d$-th components of the GV mean vectors of the natural and generated speech, respectively. They are calculated using training data.

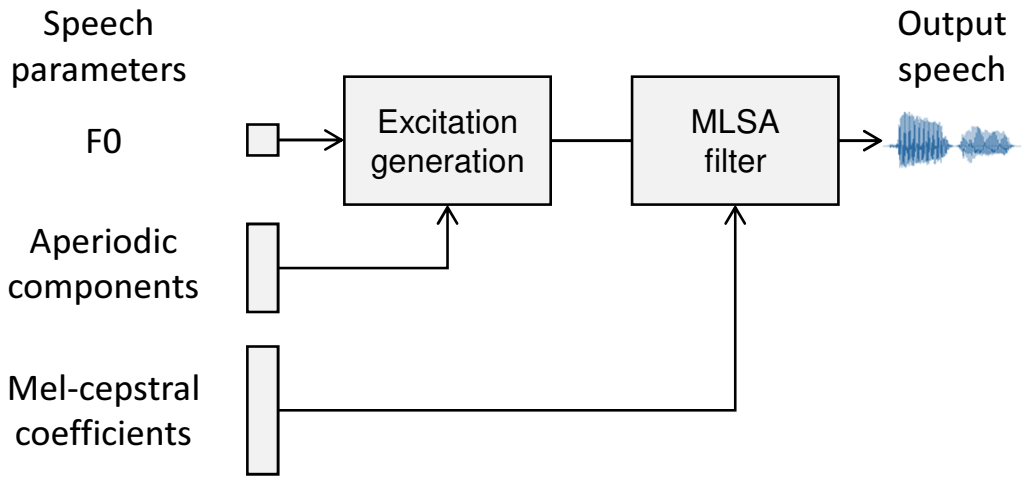


Fig. 2.8: Speech synthesis using MLSA filter. The excitation signal is firstly generated from $F_0$ and aperiodic components and then the MLSA filter is applied to the signal for synthesizing the speech waveform.

### 2.2.4 Speech Waveform Synthesis

#### Mel-log Spectrum Approximation (MLSA) Filter

The synthetic speech waveform is synthesized from the generated speech parameters, using a synthesis filter such as the mel-log spectrum approximation (MLSA) filter [42]. Figure 2.8 illustrates the speech synthesis process using the MLSA filter.

#### Spectral Differentials Filter

The analysis of the excitation parameters often occurs some errors such as the U/V decision error. In VC using spectral differentials [29], the converted speech waveform is synthesized by applying a spectral differential filter to the input speech waveform. This technique can avoid causing the errors and improve the converted speech quality. Figure 2.9 illustrates the VC process using the spectral differentials.

## 2.3 Vocoder-free Statistical Parametric Speech Synthesis

### 2.3.1 DNN-based Acoustic Models for SPSS using STFT Spectra

The conventional SPSS using vocoder-derived speech parameters works reasonably well. However, as mentioned in Section 2.2.4, use of the vocoder-based parameterization cause buzziness in the synthetic speech. One method for preventing this is vocoder-free SPSS using STFT spectra [18].

Acoustic models predict the static feature sequence of the STFT spectral amplitudes from a joint vector of the linguistic features, continuous $F_0$, and U/V. Use of the $F_0$ parameters as the input features is an effective wat to predict the harmonic information

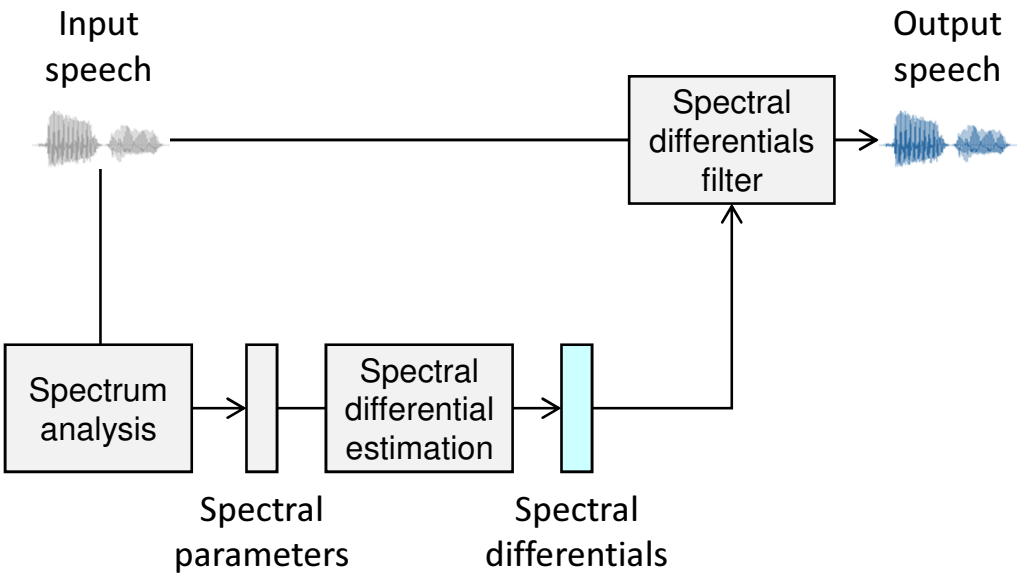


Fig. 2.9: Voice conversion using spectral differentials. The input speech waveform is converted using the estimated spectral differentials filter.

of the spectral amplitudes [18]. Therefore, $F_0$ models need to be constructed that generate the $F_0$ parameters from the linguistic features apart from the acoustic models for the STFT spectral amplitudes. The duration models used in the conventional TTS are also used.

The acoustic models are trained to minimize the MSE between the natural and generated static features of the spectral amplitudes. Although the MGE used in conventional SPSS can be used in vocoder-free SPSS, the MSE used by Takaki et al. [18] is used here.

### 2.3.2 Phase Reconstruction from Spectral Amplitudes

The phase information is reconstructed using Griffin and Lim's algorithm [43] with the predicted spectral amplitudes. Let $\mathrm{y}(n)$ be a speech waveform sample at time index $n$. The phase information for the given STFT spectral amplitudes $y_t(d)$ is reconstructed in accordance with Algorithm 2.1. Use of this algorithm enables a speech waveform to be synthesized without the vocoding process. Figure 2.10 illustrates the speech synthesis process in SPSS using STFT spectra.

## 2.4 Summary

This chapter described the basic framework of SPSS using DNNs as the acoustic models. In SPSS, speech synthesis is performed in several steps, including feature analysis, acoustic model training, speech parameter generation, and speech waveform synthesis. The DNNs described in this chapter play an important role in SPSS: they represent the relationships between the input features and speech parameters. Feed-Forward DNNs and LSTMs are often used as the DNN architecture, and their model parameters are estimated using the

**Algorithm 2.1** Phase reconstruction from spectral amplitudes

1: set initial phase information $\phi_t(d)$ to random values
2: set initial STFT spectra $\mathrm{Y}_t(d)$ to $y_t(d)\exp(j\phi_t(d))$
3: **for** number of iterations **do**
4: generate $\mathrm{y}(n)$ from $\mathrm{Y}_t(d)$ using inverse STFT (ISTFT):

$$\mathrm{y}(n) \leftarrow \mathrm{ISTFT}\left[\mathrm{Y}_t(d)\right].$$

5: reconstruct $\mathrm{Y}_t(d)$ from $\mathrm{y}(n)$ using STFT:

$$\mathrm{Y}_t(d) \leftarrow \mathrm{STFT}\left[\mathrm{y}(n)\right].$$

6: update $\mathrm{Y}_t(d)$ with fixed spectral amplitudes $y_t(d)$:

$$\mathrm{Y}_t(d) \leftarrow y_t(d)\frac{\mathrm{Y}_t(d)}{|\mathrm{Y}_t(d)|}.$$

7: **end for**

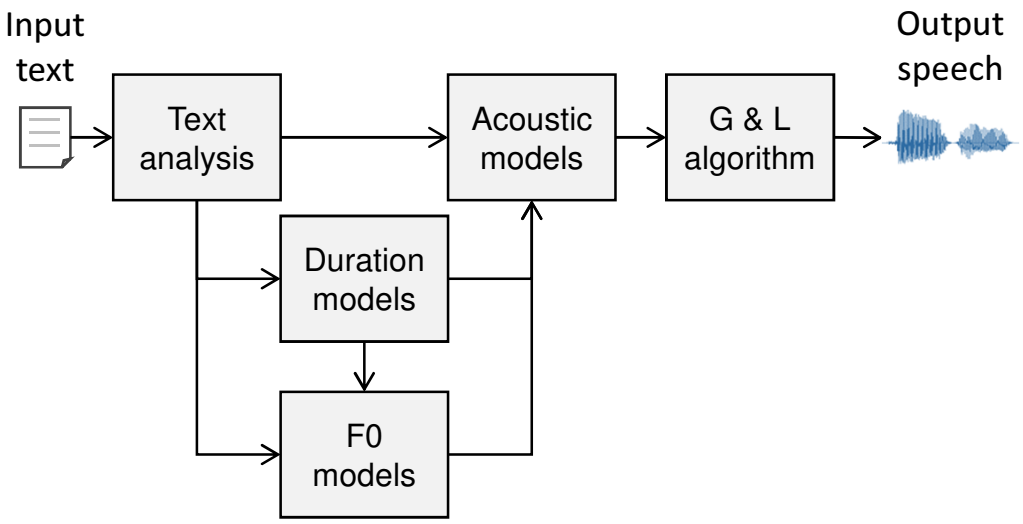


Fig. 2.10: TTS process using STFT spectra. "G & L" indicates "Griffin and Lim."

BP (or BPTT) algorithm so as to minimize the defined loss function.

# Chapter 3

# Statistical Parametric Speech Synthesis Using Generative Adversarial Networks

## 3.1 Introduction

Although a variety of methods such as the GV compensation described in Section 2.2.3 is effective for improving synthetic speech quality, the over-smoothing effect is still a critical problem in SPSS. To overcome the effect, this chapter proposes a novel algorithm for training acoustic models. In the proposed algorithm, a framework of generative adversarial networks (GANs) is incorporated into the acoustic model training. Since the objective of the GANs is to minimize the distribution difference between natural and generated samples, the proposed algorithm can reproduce natural statistics of speech parameters.

This chapter is organized as follows. Section 3.2 explains a basic framework of the GANs. Section 3.3 describes the proposed algorithm for acoustic modeling incorporating the GANs. Section 3.4 presents experimental evaluations of the proposed algorithm in DNN-based TTS and VC. Section 3.5 summarizes this chapter.

## 3.2 Generative Adversarial Networks (GANs)

### 3.2.1 Objective of GANs

GANs [44] are frameworks for learning deep generative models, which simultaneously train two DNNs: a generator and discriminator $D(\boldsymbol{y};\theta_{\mathrm{D}})$. $\theta_{\mathrm{D}}$ is a set of the model parameters of the discriminator given as neural networks. The value obtained by taking the sigmoid function from the discriminator's output, $1/(1+\exp(-D(\boldsymbol{y})))$, represents the posterior probability that input $\boldsymbol{y}$ is a natural sample. The discriminator is trained to make the posterior probability 1 for natural samples and 0 for generated samples, while the generator is trained to deceive the discriminator; that is, it tries to make the discriminator make the posterior probability 1 for generated samples. In the GAN training, the two DNNs are iteratively updated by minibatch stochastic gradient descent. Figure 3.1 illustrates a

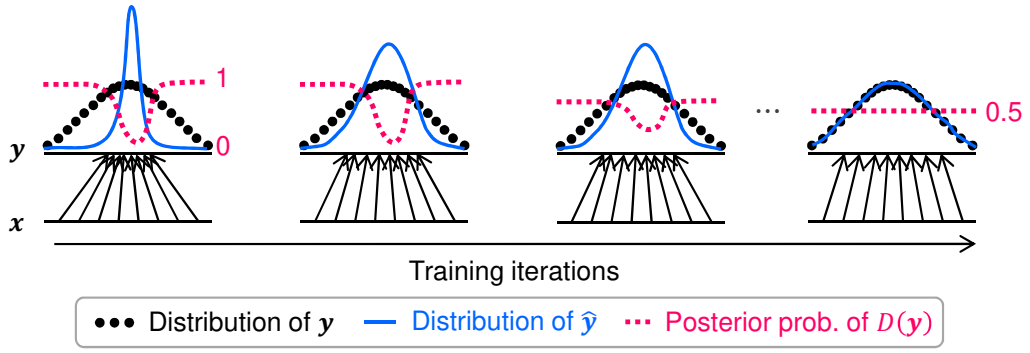


Fig. 3.1: GAN framework. The discriminator is trained to distinguish $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$, while the generator is trained to deceieve it. Here, $\hat{\boldsymbol{y}}$ is generated from $\boldsymbol{x}$ through the generator.

conceptural diagram of the GAN framework.

### 3.2.2 Discriminative Model Training

By using a natural sample $\boldsymbol{y}$ and generated sample $\hat{\boldsymbol{y}}$, we calculate the discriminator loss $L_{\mathrm{D}}^{(\mathrm{GAN})}(\boldsymbol{y}, \hat{\boldsymbol{y}})$ defined as the following cross-entropy function:

$$\begin{aligned} L_{\mathrm{D}}^{(\mathrm{GAN})}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = &-\frac{1}{T}\sum_{t=1}^{T}\log\frac{1}{1+\exp\left(-D\left(\boldsymbol{y}_t\right)\right)} \\ &-\frac{1}{T}\sum_{t=1}^{T}\log\left(1-\frac{1}{1+\exp\left(-D\left(\hat{\boldsymbol{y}}_t\right)\right)}\right). \end{aligned} \tag{3.1}$$

$\theta_{\mathrm{D}}$ is updated by using the stochastic gradient $\nabla_{\theta_{\mathrm{D}}} L_{\mathrm{D}}^{(\mathrm{GAN})}(\boldsymbol{y}, \hat{\boldsymbol{y}})$. Figure 3.2 illustrates the procedure for computing the discriminator loss.

### 3.2.3 Generative Model Training

After updating the discriminator, we calculate the adversarial loss of the generator $L_{\mathrm{ADV}}^{(\mathrm{GAN})}(\hat{\boldsymbol{y}})$ which deceives the discriminator as follows:

$$L_{\mathrm{ADV}}^{(\mathrm{GAN})}(\hat{\boldsymbol{y}}) = -\frac{1}{T}\sum_{t=1}^{T}\log\frac{1}{1+\exp\left(-D\left(\hat{\boldsymbol{y}}_t\right)\right)}. \tag{3.2}$$

A set of the model parameters of the generator $\theta_{\mathrm{G}}$ is updated by using the stochastic gradient $\nabla_{\theta_{\mathrm{G}}} L_{\mathrm{ADV}}^{(\mathrm{GAN})}(\hat{\boldsymbol{y}})$. Goodfellow et al. [44] showed this adversarial framework minimizes the approximated Jensen–Shannon (JS) divergence between two distributions of natural and generated samples.

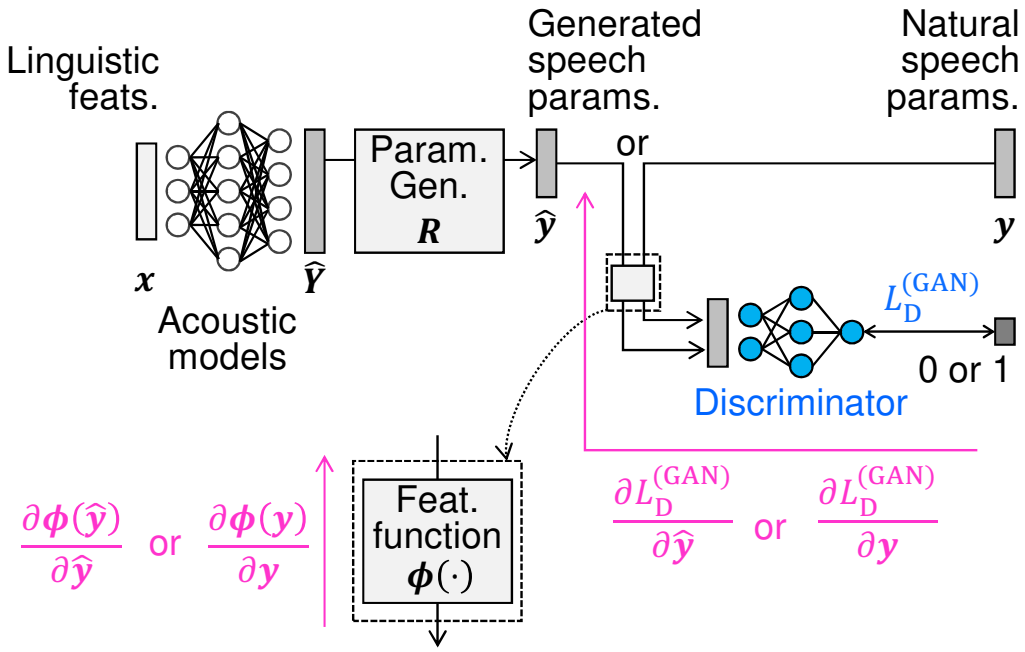


Fig. 3.2: Loss function and gradients for updating the discriminator. Param. Gen. indicates MLPG [1]. Note that, the model parameters of the acoustic models are not updated in this step.

## 3.3 Acoustic Model Training Using GANs

### 3.3.1 Acoustic Model Training Criteria Incorporating GANs

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 the acoustic model training is defined as the following:

$$L_{\mathrm{G}}\left(\boldsymbol{y}, \hat{\boldsymbol{y}}\right) = L_{\mathrm{MGE}}\left(\boldsymbol{y}, \hat{\boldsymbol{y}}\right) + \omega_{\mathrm{D}} \frac{E_{L_{\mathrm{MGE}}}}{E_{L_{\mathrm{ADV}}}} L_{\mathrm{ADV}}^{(\mathrm{GAN})}\left(\hat{\boldsymbol{y}}\right), \tag{3.3}$$

where $L_{\mathrm{ADV}}^{(\mathrm{GAN})}\left(\hat{\boldsymbol{y}}\right)$ 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. $E_{L_{\mathrm{MGE}}}$ and $E_{L_{\mathrm{ADV}}}$ denote the expectation values of $L_{\mathrm{MGE}}(\boldsymbol{y}, \hat{\boldsymbol{y}})$ and $L_{\mathrm{ADV}}^{(\mathrm{GAN})}\left(\hat{\boldsymbol{y}}\right)$, respectively. Their ratio $E_{L_{\mathrm{MGE}}}/E_{L_{\mathrm{ADV}}}$ is the scale normalization term between the two loss functions, and the hyper-parameter $\omega_{\mathrm{D}}$ controls the weight of the second term. When $\omega_{\mathrm{D}} = 0$, the loss function is equivalent to the conventional MGE training described in Section 2.2.2, and when $\omega_{\mathrm{D}} = 1$, the two loss functions have equal weights. A set of the model parameters of the acoustic models $\theta_{\mathrm{G}}$ is updated by using

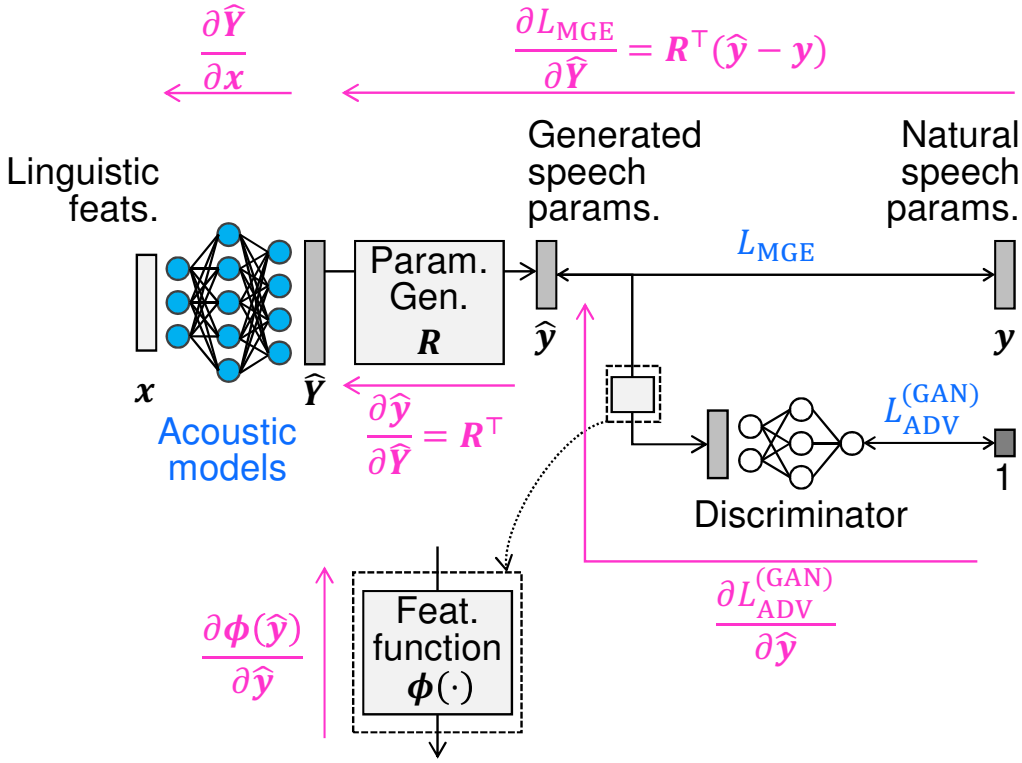


Fig. 3.3: Loss functions and gradients for updating acoustic models in the proposed method. Note that the model parameters of the discriminator are not updated in this step.

the stochastic gradient $\nabla_{\theta_{\mathrm{G}}} L_{\mathrm{G}}(\hat{\boldsymbol{y}})$. Figure 3.3 illustrates the procedure for computing the proposed loss function. In our algorithm, the acoustic models and discriminator are iteratively optimized, as shown in Algorithm 3.1. 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 $L_{\mathrm{ADV}}^{(\mathrm{GAN})}(\hat{\boldsymbol{y}})$ in $L_{\mathrm{G}}(\boldsymbol{y}, \hat{\boldsymbol{y}})$, $\theta_{\mathrm{D}}$ is not updated by the BP algorithm for the acoustic models.

### 3.3.2 Integrating Anti-spoofing Techniques

The discriminator used in our method can be regarded as a DNN-based anti-spoofing (voice spoofing detection) [45, 46] that distinguishes natural and synthetic speech. From this perspective, a feature function $\boldsymbol{\phi}(\cdot)$ can be inserted between speech parameter prediction and the discriminator as shown in Figs. 3.2 and 3.3. The function calculates more distinguishable features in anti-spoofing than the direct use of speech parameters themselves. Namely, instead of $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$ in Eqs. (3.1) and (3.2), $\boldsymbol{\phi}(\boldsymbol{y})$ and $\boldsymbol{\phi}(\hat{\boldsymbol{y}})$ are used. In training the acoustic models, the gradient $\partial \boldsymbol{\phi}(\hat{\boldsymbol{y}}) / \partial \hat{\boldsymbol{y}}$ is used for the BP algorithm.

As the features that are effective in anti-spoofing, this thesis uses dynamic features of the spectral parameters, which are more effective to detect synthetic speech than the

**Algorithm 3.1** Iterative optimization for acoustic models and discriminator

1: $\eta :=$ learning rate
2: **for** number of training iterations **do**
3: **for all** training data $(\boldsymbol{x}, \boldsymbol{y})$ **do**
4: generate $\hat{\boldsymbol{y}}$ from the acoustic models:

$$\hat{\boldsymbol{y}} = \boldsymbol{G}(\boldsymbol{x}; \theta_{\mathrm{G}}).$$

5: update $\theta_{\mathrm{D}}$ while fixing $\theta_{\mathrm{G}}$:

$$\theta_{\mathrm{D}} \leftarrow \theta_{\mathrm{D}} - \eta \boldsymbol{\nabla}_{\theta_{\mathrm{D}}} L_{\mathrm{D}}^{(\mathrm{GAN})}(\boldsymbol{y}, \hat{\boldsymbol{y}}).$$

6: update $\theta_{\mathrm{G}}$ while fixing $\theta_{\mathrm{D}}$:

$$\theta_{\mathrm{G}} \leftarrow \theta_{\mathrm{G}} - \eta \boldsymbol{\nabla}_{\theta_{\mathrm{G}}} L_{\mathrm{G}}(\boldsymbol{y}, \hat{\boldsymbol{y}}).$$

7: **end for**
8: **end for**

directly use of static features [47]. The feature function is defined as $\boldsymbol{\phi}(\hat{\boldsymbol{y}}) = \boldsymbol{M}\hat{\boldsymbol{y}}$, and the gradient $\boldsymbol{M}^{\top}$ is used for the BP algorithm.

Besides the dynamic features, we can use many features to be incorporated into the proposed training. Because the vocoder systems are based on a minimum-phase vocal tract model, the differences between the phase spectra between natural and synthetic speech can be utilized for the discrimination [48]. Based on the difficulty in reliable prosody modelling, features related to the $F_0$ statistics are also effective to detect spoofing attacks [49, 50]. To capture long-term dependencies of speech parameters, temporal magnitude/phase modulation features were proposed in [51].

### 3.3.3 Duration Model Training Considering Isochrony

Our algorithm is simply applied to the spectral parameters and continuous $F_0$ generation. Here, we extend our algorithm to duration generation in TTS. For duration generation, although we can directly apply our algorithm to phoneme duration, it is not guaranteed that naturally-distributed phoneme duration has natural isochrony of the target language (e.g., moras in Japanese) [52]. Therefore, we modify our algorithm so that the generated duration naturally distributes in the language-dependent isochrony level. Figure 3.4 shows the architecture. In the case of Japanese, which has mora isochrony, each mora duration is calculated from the corresponding phoneme durations. The discriminator 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. 3.5, the

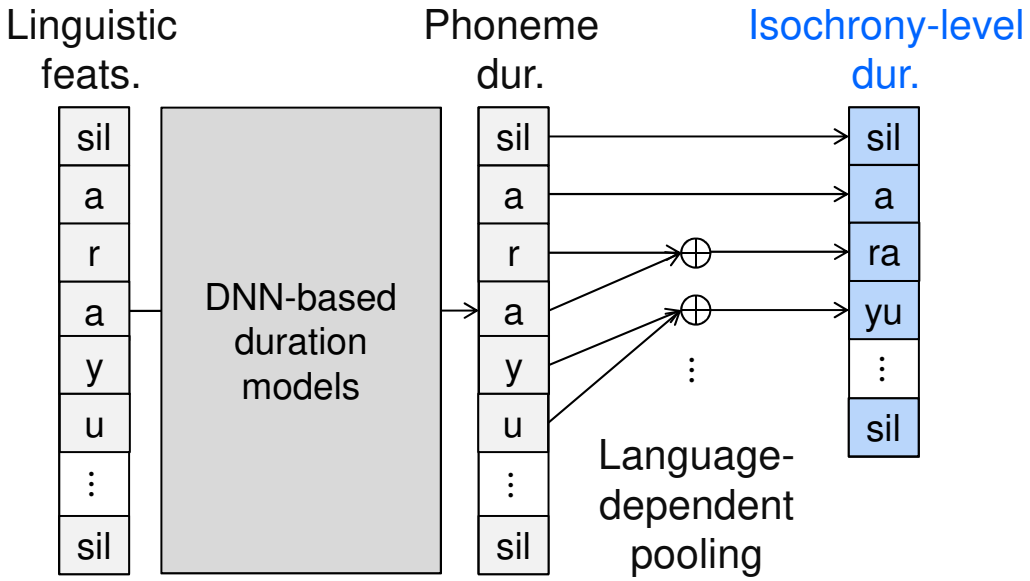


Fig. 3.4: Architecture to calculate isochrony-level duration from phoneme duration. In the case of Japanese, which has mora isochrony, each mora duration is calculated from the corresponding phoneme duration, e.g., the mora duration of /ra/ is calculated as the sum of the phoneme durations of /r/ and /a/.

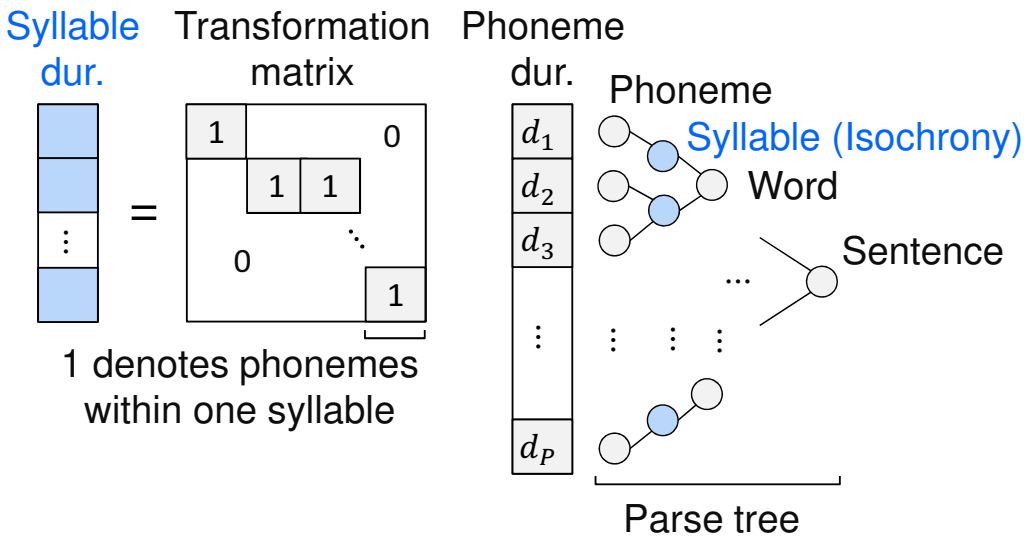


Fig. 3.5: Matrix representation to calculate isochrony-level duration. This is an example in the case of a syllable-timed language such as Chinese.

BP algorithm is done using the transpose of the transformation matrix.

### 3.3.4 Various Divergences Miminized by GANs

The GAN framework works as a divergence minimization between natural and generated speech parameters. As described in Section 3.2, the original GAN [44] minimizes the

approximated JS divergence. From the perspective of the divergence minimization, we further introduce additional GANs minimizing other divergences: $f$-GAN [53], Wasserstein GAN (W-GAN) [54], and least squares GAN (LS-GAN) [55]. The divergence of the $f$-GAN is strongly related to speech processing such as a nonnegative matrix factorization [56, 57], and the effectiveness of the W-GAN and LS-GAN in the image processing is known. The discriminator loss $L_{\mathrm{D}}^{(*\text{-GAN})}(\boldsymbol{y}, \hat{\boldsymbol{y}})$ and adversarial loss $L_{\mathrm{ADV}}^{(*\text{-GAN})}(\hat{\boldsymbol{y}})$ introduced below can be used instead of Eqs. (3.1) and (3.2), respectively.

### $f$-GAN

The $f$-GAN [53] is the unified framework that encompasses the original GAN. The difference between distributions of natural and generated data is defined as the $f$-divergence [58], which is a large class of different divergences including the Kullback–Leibler (KL) and JS divergence. The $f$-divergence $\mathcal{D}_f(\boldsymbol{y}\|\hat{\boldsymbol{y}})$ is defined as follows:

$$\mathcal{D}_f\left(\boldsymbol{y}\|\hat{\boldsymbol{y}}\right) = \int q\left(\hat{\boldsymbol{y}}\right) f\left(\frac{p\left(\boldsymbol{y}\right)}{q\left(\hat{\boldsymbol{y}}\right)}\right) d\boldsymbol{y}, \tag{3.4}$$

where $p(\cdot)$ and $q(\cdot)$ are absolutely continuous density functions of $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$, 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:

$$\mathcal{D}_{\mathrm{KL}}\left(\boldsymbol{y}\|\hat{\boldsymbol{y}}\right) = \int p\left(\boldsymbol{y}\right) \log \frac{p\left(\boldsymbol{y}\right)}{q\left(\hat{\boldsymbol{y}}\right)} d\boldsymbol{y}. \tag{3.5}$$

The discriminator loss $L_{\mathrm{D}}^{(\text{KL-GAN})}\left(\boldsymbol{y}, \hat{\boldsymbol{y}}\right)$ is defined as follows:

$$\begin{aligned} L_{\mathrm{D}}^{(\text{KL-GAN})}\left(\boldsymbol{y}, \hat{\boldsymbol{y}}\right) = &-\frac{1}{T}\sum_{t=1}^{T} D\left(\boldsymbol{y}_t\right) \\ &+ \frac{1}{T}\sum_{t=1}^{T} \exp\left(D\left(\hat{\boldsymbol{y}}_t\right) - 1\right), \end{aligned} \tag{3.6}$$

while the adversarial loss $L_{\mathrm{ADV}}^{(\text{KL-GAN})}\left(\hat{\boldsymbol{y}}\right)$ is defined as follows:

$$L_{\mathrm{ADV}}^{(\text{KL-GAN})}\left(\hat{\boldsymbol{y}}\right) = -\frac{1}{T}\sum_{t=1}^{T} D\left(\hat{\boldsymbol{y}}_t\right). \tag{3.7}$$

**Reversed KL (RKL)-GAN:** Since the KL divergence is not symmetric, the reversed version, called reversed KL (RKL) divergence $\mathcal{D}_{\mathrm{RKL}}(\boldsymbol{y}\|\hat{\boldsymbol{y}})$ differs from $\mathcal{D}_{\mathrm{KL}}(\boldsymbol{y}\|\hat{\boldsymbol{y}})$, which is defined as follows:

$$\mathcal{D}_{\mathrm{RKL}}\left(\boldsymbol{y}\|\hat{\boldsymbol{y}}\right) = \int q\left(\hat{\boldsymbol{y}}\right) \log \frac{q\left(\hat{\boldsymbol{y}}\right)}{p\left(\boldsymbol{y}\right)} d\boldsymbol{y} = \mathcal{D}_{\mathrm{KL}}\left(\hat{\boldsymbol{y}}\|\boldsymbol{y}\right). \tag{3.8}$$

Defining $f(r) = -\log r$ gives the discriminator loss $L_{\mathrm{D}}^{(\text{RKL-GAN})}(\boldsymbol{y}, \hat{\boldsymbol{y}})$ as follows:

$$\begin{aligned} L_{\mathrm{D}}^{(\text{RKL-GAN})}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{T}\sum_{t=1}^{T}\exp\left(-D\left(\boldsymbol{y}_t\right)\right) \\ + \frac{1}{T}\sum_{t=1}^{T}\left(-1 + D\left(\hat{\boldsymbol{y}}_t\right)\right), \end{aligned} \tag{3.9}$$

while the adversarial loss $L_{\mathrm{ADV}}^{(\text{RKL-GAN})}(\hat{\boldsymbol{y}})$ is defined as follows:

$$L_{\mathrm{ADV}}^{(\text{RKL-GAN})}(\hat{\boldsymbol{y}}) = \frac{1}{T}\sum_{t=1}^{T}\exp\left(-D\left(\hat{\boldsymbol{y}}_t\right)\right). \tag{3.10}$$

**JS-GAN:** The JS divergence without approximation can be formed within the $f$-GAN framework. Defining $f(r) = -(r+1)\log\frac{r+1}{2} + r\log r$ gives the JS divergence as follows:

$$\begin{aligned} \mathcal{D}_{\mathrm{JS}}\left(\boldsymbol{y}\|\hat{\boldsymbol{y}}\right) = \frac{1}{2}\int p\left(\boldsymbol{y}\right)\log\frac{2p\left(\boldsymbol{y}\right)}{p\left(\boldsymbol{y}\right)+q\left(\hat{\boldsymbol{y}}\right)}d\boldsymbol{y} \\ + \frac{1}{2}\int q\left(\hat{\boldsymbol{y}}\right)\log\frac{2q\left(\hat{\boldsymbol{y}}\right)}{p\left(\boldsymbol{y}\right)+q\left(\hat{\boldsymbol{y}}\right)}d\boldsymbol{y}. \end{aligned} \tag{3.11}$$

the discriminator loss $L_{\mathrm{D}}^{(\text{JS-GAN})}(\boldsymbol{y}, \hat{\boldsymbol{y}})$ is defined as follows:

$$\begin{aligned} L_{\mathrm{D}}^{(\text{JS-GAN})}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = -\frac{1}{T}\sum_{t=1}^{T}\log\frac{2}{1+\exp\left(-D\left(\boldsymbol{y}_t\right)\right)} \\ - \frac{1}{T}\sum_{t=1}^{T}\log\left(2 - \frac{2}{1+\exp\left(-D\left(\hat{\boldsymbol{y}}_t\right)\right)}\right), \end{aligned} \tag{3.12}$$

while the adversarial loss $L_{\mathrm{ADV}}^{(\text{JS-GAN})}(\hat{\boldsymbol{y}})$ is defined as follows:

$$L_{\mathrm{ADV}}^{(\text{JS-GAN})}(\hat{\boldsymbol{y}}) = -\frac{1}{T}\sum_{t=1}^{T}\log\frac{2}{1+\exp\left(-D\left(\hat{\boldsymbol{y}}_t\right)\right)}. \tag{3.13}$$

Note that, the approximated JS divergence minimized by the original GAN is $2\mathcal{D}_{\mathrm{JS}}(\boldsymbol{y}\|\hat{\boldsymbol{y}}) - \log(4)$ [44].

### Wasserstein GAN (W-GAN)

To stabilize the extremely unstable training of the original GAN, Arjovsky et al. [54] proposed the W-GAN, which minimizes the Earth-Mover's distance (Wasserstein-1). The Earth-Mover's distance is defined as follows:

$$\mathcal{D}_{\mathrm{EM}}\left(\boldsymbol{y}, \hat{\boldsymbol{y}}\right) = \inf_{\gamma}\mathbb{E}_{(\boldsymbol{y},\hat{\boldsymbol{y}})\sim\gamma}\left[\|\boldsymbol{y} - \hat{\boldsymbol{y}}\|\right], \tag{3.14}$$

where $\gamma(\boldsymbol{y}, \hat{\boldsymbol{y}})$ is the joint distribution whose marginals are respectively the distributions of $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$. On the basis of the Kantorovich–Rubinstein duality [59], the discriminator

loss $L_{\mathrm{D}}^{(\text{W-GAN})}(\boldsymbol{y}, \hat{\boldsymbol{y}})$ is defined as follows:

$$L_{\mathrm{D}}^{(\text{W-GAN})}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = -\frac{1}{T}\sum_{t=1}^{T} D(\boldsymbol{y}_t) + \frac{1}{T}\sum_{t=1}^{T} D(\hat{\boldsymbol{y}}_t), \tag{3.15}$$

while the adversarial loss $L_{\mathrm{ADV}}^{(\text{W-GAN})}(\hat{\boldsymbol{y}})$ is defined as follows:

$$L_{\mathrm{ADV}}^{(\text{W-GAN})}(\hat{\boldsymbol{y}}) = -\frac{1}{T}\sum_{t=1}^{T} D(\hat{\boldsymbol{y}}_t). \tag{3.16}$$

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 $[-0.01, 0.01]$.

#### Least Squares GAN (LS-GAN)

To avoid the gradient vanishing problem of the original GAN using the sigmoid cross entropy, Mao et al. [55] proposed the LS-GAN, which formulates the objective function minimizing the mean squared error. The discriminator loss $L_{\mathrm{D}}^{(\text{LS-GAN})}(\boldsymbol{y}, \hat{\boldsymbol{y}})$ is defined as follows:

$$\begin{aligned} L_{\mathrm{D}}^{(\text{LS-GAN})}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = &\frac{1}{2T}\sum_{t=1}^{T} \left(D(\boldsymbol{y}_t) - b\right)^2 \\ &+ \frac{1}{2T}\sum_{t=1}^{T} \left(D(\hat{\boldsymbol{y}}_t) - a\right)^2, \end{aligned} \tag{3.17}$$

while the adversarial loss $L_{\mathrm{ADV}}^{(\text{LS-GAN})}(\hat{\boldsymbol{y}})$ is defined as follows:

$$L_{\mathrm{ADV}}^{(\text{LS-GAN})}(\hat{\boldsymbol{y}}) = \frac{1}{2T}\sum_{t=1}^{T} \left(D(\hat{\boldsymbol{y}}_t) - c\right)^2, \tag{3.18}$$

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 $\mathcal{X}^2$ divergence between $p(\boldsymbol{y}) + q(\hat{\boldsymbol{y}})$ and $2q(\hat{\boldsymbol{y}})$. Because we found that these conditions degrade quality of synthetic speech, we used alternative conditions suggested in Eq. (9) of [55], i.e., $a = 0$, $b = 1$, and $c = 1$.

### 3.3.5 Discussions

The proposed loss function (Eq. (3.3)) is the combination of a multi-task learning algorithm using discriminators [60] and GANs. In defining $L_{\mathrm{G}}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = L_{\mathrm{ADV}}^{(\text{GAN})}(\hat{\boldsymbol{y}})$, 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 [61] without a latent variable. Also, since only the BP algorithm is used for training, a variety of DNN architectures such as long short-term memory (LSTM) [62] can be used as the acoustic models and discriminator.

Table 3.1: Statistics of natural ("Natural") and generated ("MGE" and "Proposed") continuous $F_0$

| | Mean | Variance |
|---|---|---|
| Natural | 4.8784 | 0.076853 |
| MGE | 4.8388 | 0.032841 |
| Proposed ($\omega_{\mathrm{D}} = 1.0$) | **4.8410** | **0.032968** |

Using the designed feature function $\boldsymbol{\phi}(\cdot)$, we can choose not only analytically derived features (e.g., GV and MS) but also automatically derived features (e.g., auto-encoded features [63]). Note that, using the features that effectively detect synthetic speech for the proposed training algorithm does not necessarily improve synthetic speech quality, that is, the differences in these features do not always relate to the human perception in speech.

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 3.6 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 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 [64]) the algorithm changes. Figure 3.7 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 [65, 66]. Then, we calculated a maximal information coefficient (MIC) [67] to quantify a nonlinear correlation among the speech parameters. The results are shown in Fig. 3.8. As reported in [68], 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 $F_0$, phoneme duration, and mora duration are listed in Tables 3.1, 3.2, and 3.3, respectively. The bold values are the closest to natural statistics in the results. In Tables 3.2 and 3.3, "Proposed (phoneme)" and

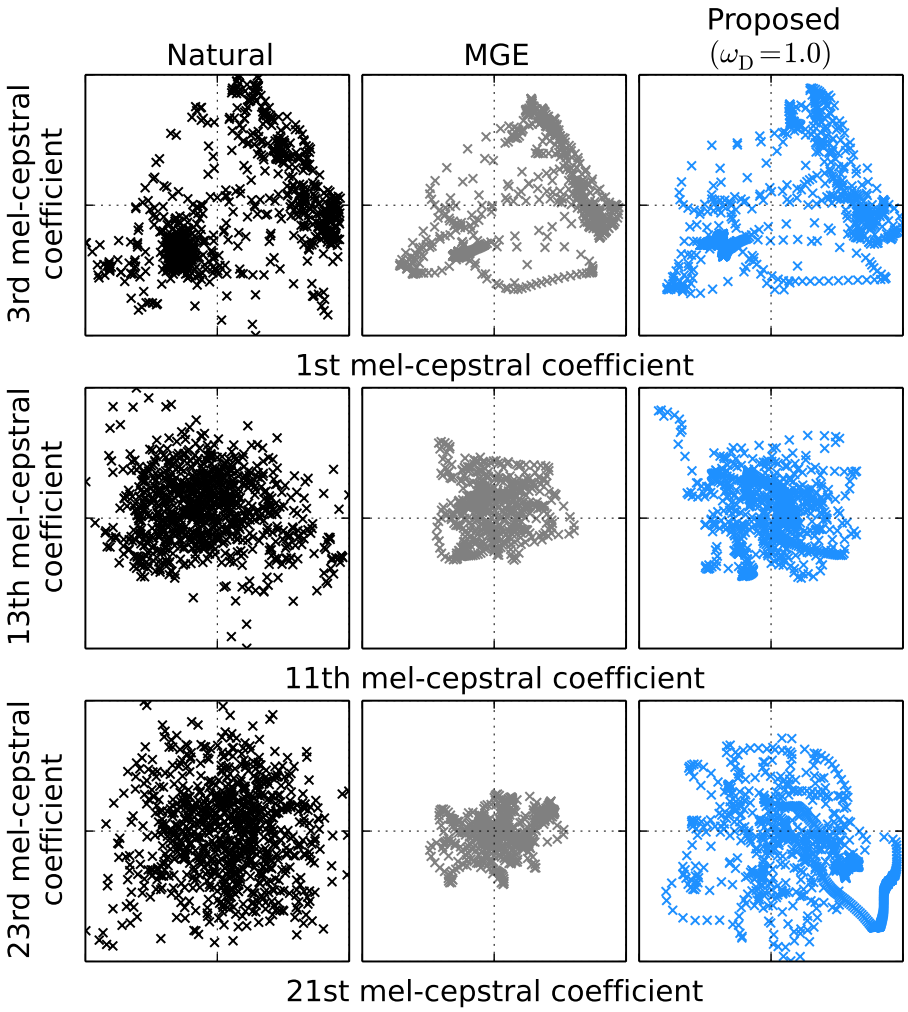


Fig. 3.6: Scatter plots of mel-cepstral coefficients with several pairs of dimensions. From the left, the figures correspond to natural speech, the conventional MGE algorithm, and the proposed algorithm ($\omega_D = 1.0$). These mel-cepstral coefficients were extracted from one utterance of the evaluation data.

Table 3.2: Statistics of natural ("Natural") and generated ("MSE" and "Proposed(*)") phoneme duration

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

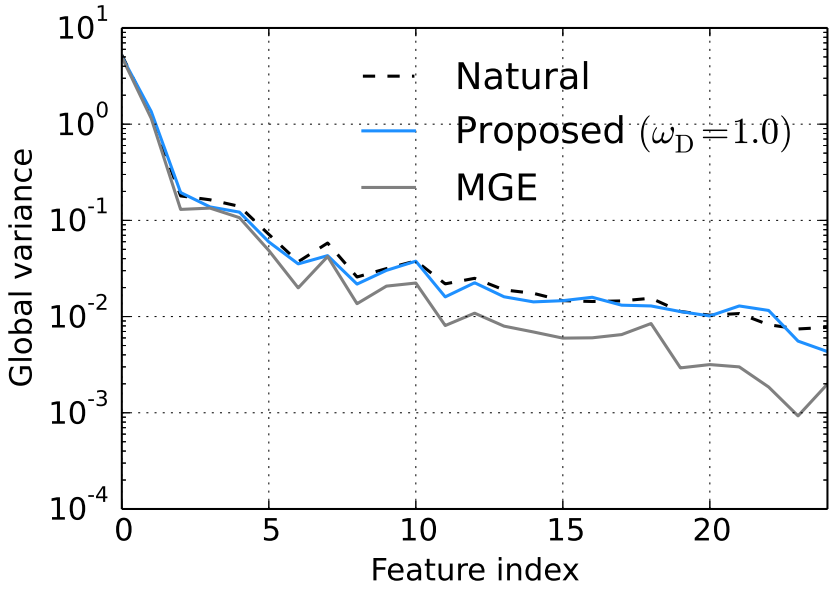


Fig. 3.7: Averaged GVs of mel-cepstral coefficients. Dashed, black, and blue lines correspond to natural speech, the conventional MGE, and the proposed algorithm, respectively.

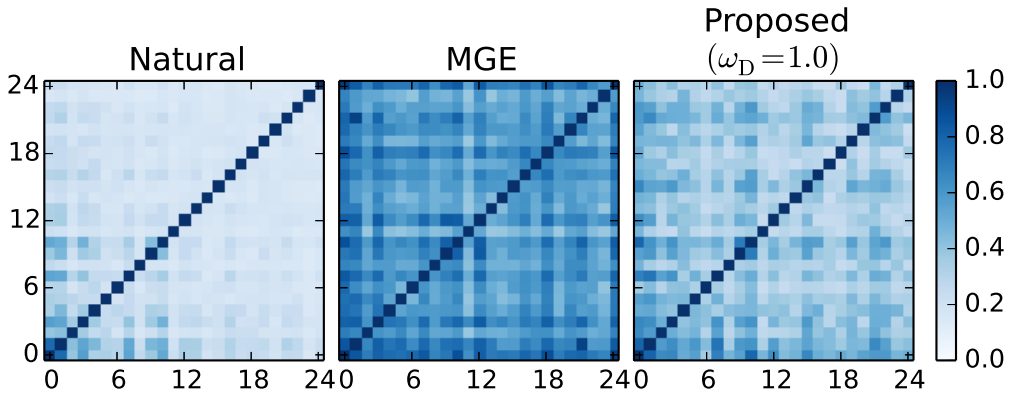


Fig. 3.8: MICs of natural and generated mel-cepstral coefficients. The MIC ranges from 0.0 to 1.0, and the two variables with a strong correlation have a value closer to 1.0. From the left, the figures correspond to natural speech, the conventional MGE algorithm, and the proposed algorithm ($\omega_D = 1.0$). These MICs were calculated from one utterance of the evaluation data.

"Proposed (mora)" indicate that the proposed methods applied 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 reduce the difference in the mean rather than in the variance.

Table 3.3: Statistics of natural ("Natural") and generated ("MSE" and "Proposed(*)") mora duration

| | Mean | Variance |
|---|---|---|
| Natural | 25.141 | 131.93 |
| MSE | 23.492 | 60.891 |
| Proposed (phoneme, $\omega_{\mathrm{D}} = 1.0$) | 24.794 | **96.828** |
| Proposed (mora, $\omega_{\mathrm{D}} = 1.0$) | **24.978** | 96.682 |

Our algorithm for spectrum and $F_0$, proposed in Section 3.3.1, compensates the joint distribution of them. Therefore, we can perform the distribution compensation considering correlations [69] between different features. Also, compensating dimensionality differences [70] 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 [71] and hierarchical duration modeling [72].

Regarding related work, Kaneko et al. [73] 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 [74]. Also, we expect that our algorithm can be extended to waveform synthesis [75, 19].

## 3.4 Experimental Evaluations for TTS

### 3.4.1 Conditions for TTS Evaluation

We used speech data of a male speaker taken from the ATR Japanese speech database [76]. 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 [21, 77] were used as excitation parameters. The STRAIGHT analysis-synthesis system [24] was used for the parameter extraction and the waveform synthesis. To improve training accuracy, speech parameter trajectory smoothing [78] 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 3.4. For the hidden layers, ReLU [35] was

Table 3.4: Architectures of DNNs used in TTS evaluations. Feed-Forward networks were used for all architectures

| | Spectral parameter generation (through Sections 3.4.2 and 3.4.4) | Spectral and $F_0$ parameter generation (Section 3.4.5) | Duration generation (Section 3.4.6) |
|---|---|---|---|
| Acoustic models | 274–3 × 400–75 | 442–3 × 512–94 | 442–3 × 512–94 |
| Discriminator | 25–2 × 200–1 | 26–3 × 256–1 | 1–3 × 256–1 |
| Duration models | N/A | 439–3 × 256–1 | 439–3 × 256–1 |

adopted to the activation function. The linear function was used for the output activation function of the acoustic models and duration models. The sigmoid function was used for the output activation function of the discriminator. In the spectral parameter generation (Section 3.4.2 through 3.4.4), 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 frame-wise static mel-cepstral coefficients (25-dim.). Here, since $F_0$, band-aperiodicity, and duration of natural speech were directly used for the speech waveform synthesis, we only used some of the prosody-related features such as the accent type. In the spectral parameter and $F_0$ generation (Section 3.4.5), the acoustic models predicted static-dynamic feature sequence of the mel-cepstral coefficients, continuous log $F_0$ [23], and band-aperiodicity with U/V (94-dim.) from the 442-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 3-4-6), 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. (2.3)) [31] frame-by-frame for the initialization of acoustic models and then we ran the conventional MGE training [39] 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 retraining were performed with 25 iterations. The expectation values $E_{L_{\mathrm{MGE}}}$ and $E_{L_{\mathrm{ADV}}}$ were estimated at each iteration step.

### 3.4.2 Objective Evaluation of Spectral Parameter Generation

In order to evaluate our algorithm, we calculated the parameter generation loss defined in Eq. (2.4) and the spoofing rate of the synthetic speech. The spoofing rate is the number of spoofing synthetic 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 $\omega_{\mathrm{D}}$ settings.

Figure 3.9 shows the results for the generation loss and spoofing rate. As $\omega_{\mathrm{D}}$ 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 $\omega_{\mathrm{D}}$ 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.

### 3.4.3 Investigation of Convergence

To investigate the convergence of the proposed training algorithm, we ran the algorithm through 100 iterations. Figure 3.10 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.

### 3.4.4 Subjective Evaluation of Spectral Parameter Generation

Preference AB tests were conducted to evaluate the quality of speech produced by the algorithm. We generated speech samples with three methods:

MGE: conventional MGE (= Proposed ($\omega_{\mathrm{D}} = 0.0$))
Proposed ($\omega_{\mathrm{D}} = 0.3$): spoofing rate $> 0.99$
Proposed ($\omega_{\mathrm{D}} = 1.0$): 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 crowd-sourced subjective evaluation systems.

The results are shown in Fig. 3.11. In Figs. 3.11(a) and (b), the proposed algorithm out-

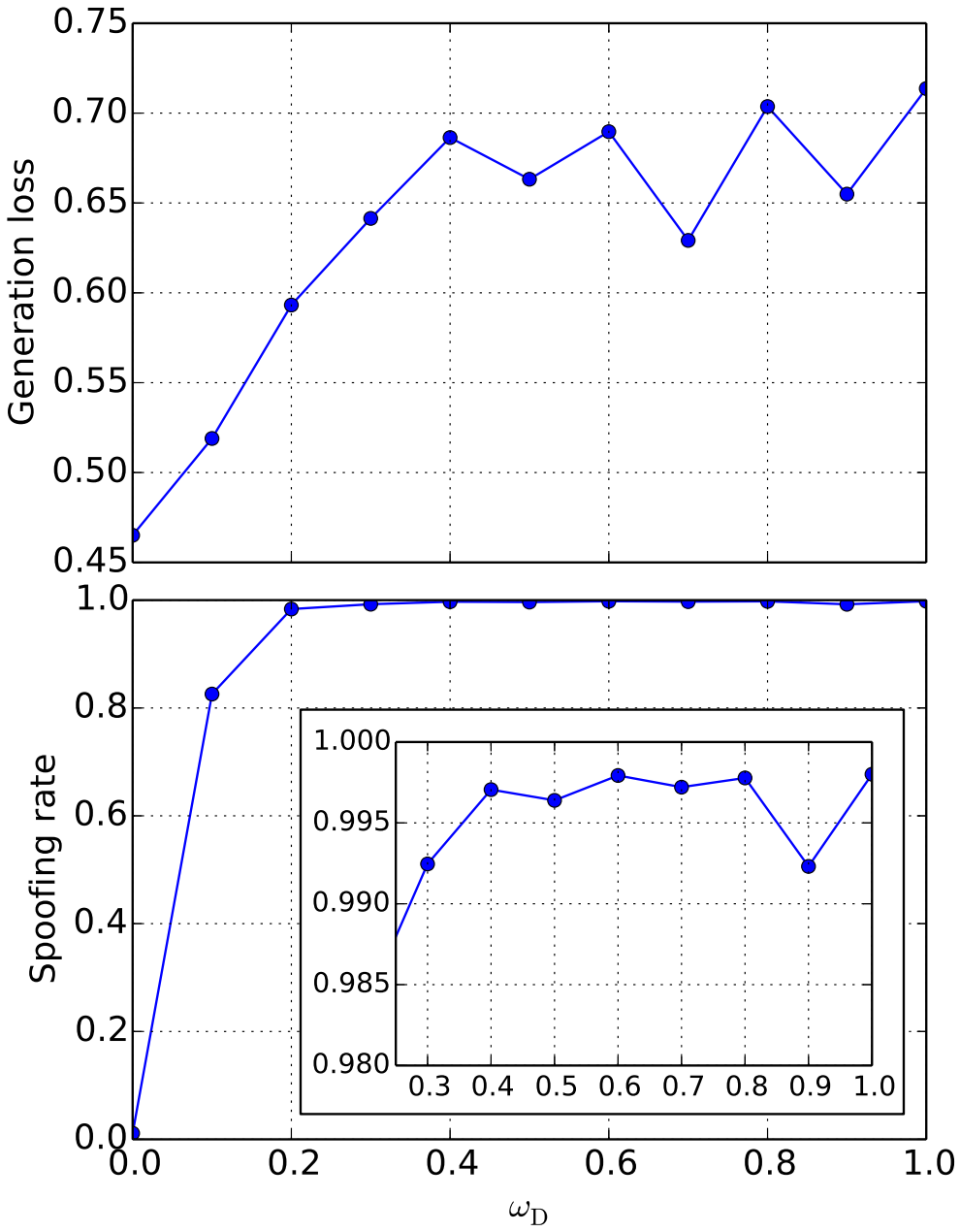


Fig. 3.9: Parameter generation loss (above) and spoofing rate (below) for various $\omega_{\mathrm{D}}$ for spectral parameter generation in TTS.

performs 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. 3.11(c) shows that the score of "Proposed ($\omega_{\mathrm{D}} = 1.0$)" was slightly better than that of "Proposed ($\omega_{\mathrm{D}} = 0.3$)."

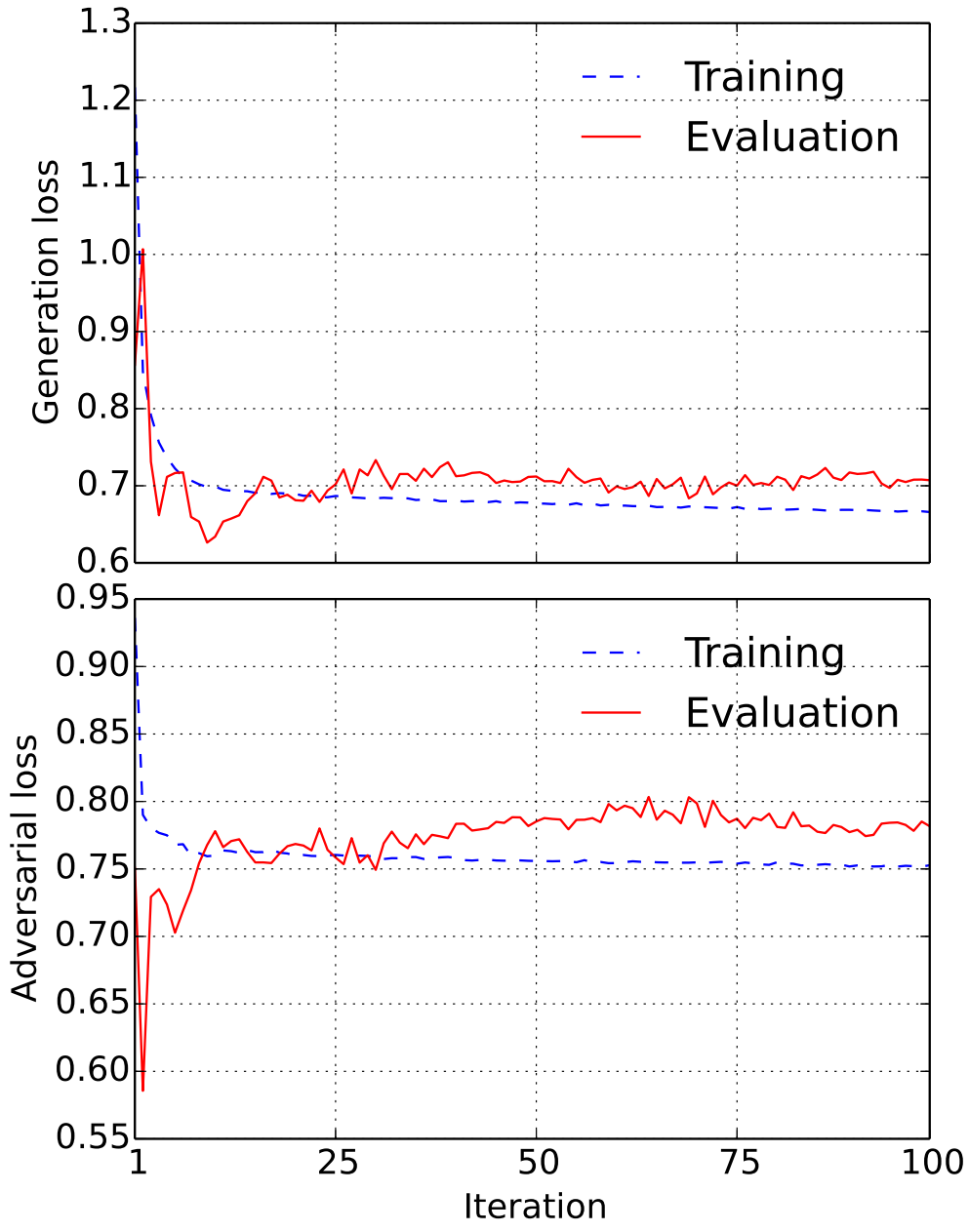


Fig. 3.10: Parameter generation loss (above) and adversarial loss (below) for the training data (blue-dashed line) and evaluation data (red line).

### 3.4.5 Subjective Evaluation of $F_0$ Generation

We evaluated the effect of the proposed algorithm for $F_0$ generation. We conducted a subjective evaluation using the following three methods:

MGE: conventional MGE

Proposed (sp): proposed algorithm applied only to spectral parameters

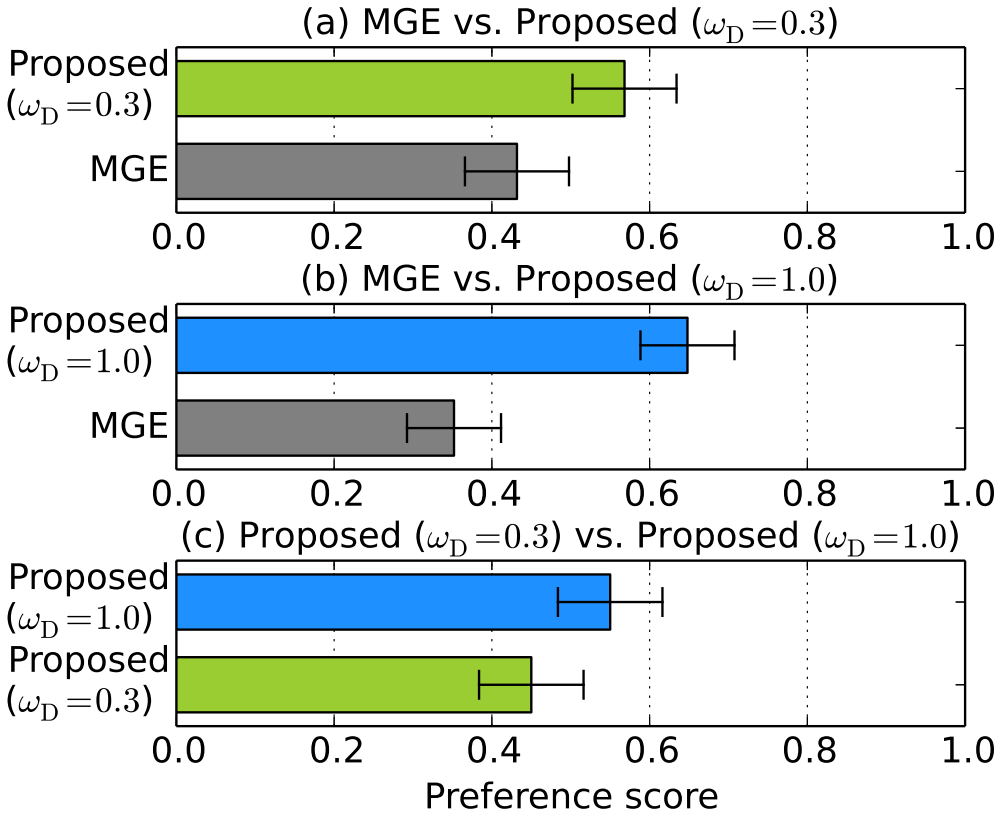


Fig. 3.11: Preference scores of speech quality with 95% confidence intervals (spectral parameter generation in TTS). From the top, the numbers of listeners were 22, 24, and 22, respectively.

Proposed (sp+F0): proposed algorithm applied to spectral and $F_0$ parameters

Every pair of synthetic speech samples generated by using each method was presented to listeners in random order. Since Fig. 3.11 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 3.12 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 $F_0$.

### 3.4.6 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

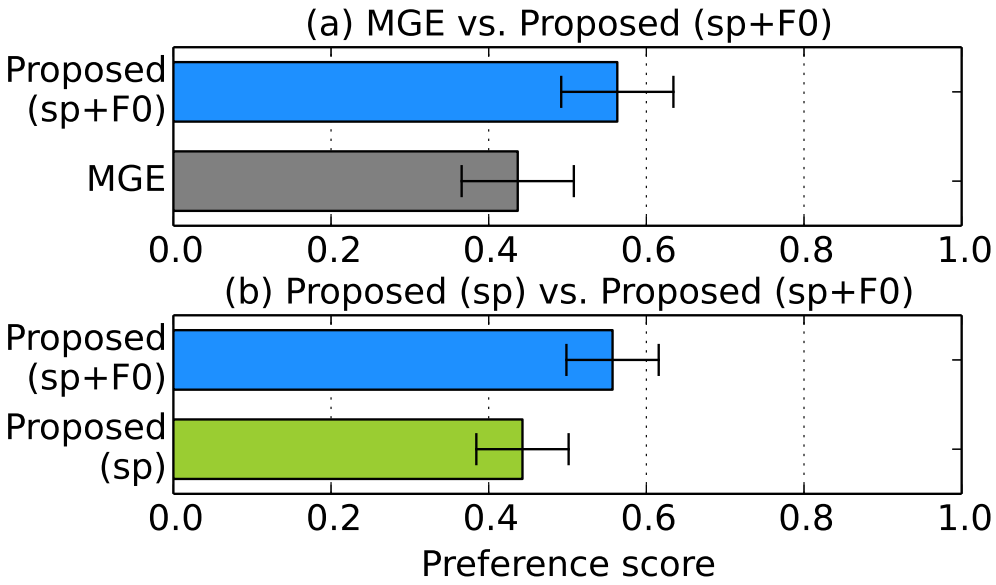


Fig. 3.12: Preference scores of speech quality with 95% confidence intervals (spectral parameter and $F_0$ generation in TTS). From the top, the numbers of the listeners were 19 and 28, respectively.

Proposed (phoneme): proposed algorithm applied to phoneme duration
Proposed (mora): proposed algorithm applied to mora duration

Preference AB tests were conducted in the same manner as in the previous evaluation.

The results are shown in Fig. 3.13. There are no significant differences in the resulting scores. To investigate the reason, we constructed an 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. 3.14, 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.

### 3.4.7 Comparison to Global Variance Compensation

Figure 3.7 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 [41] 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 3.15 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

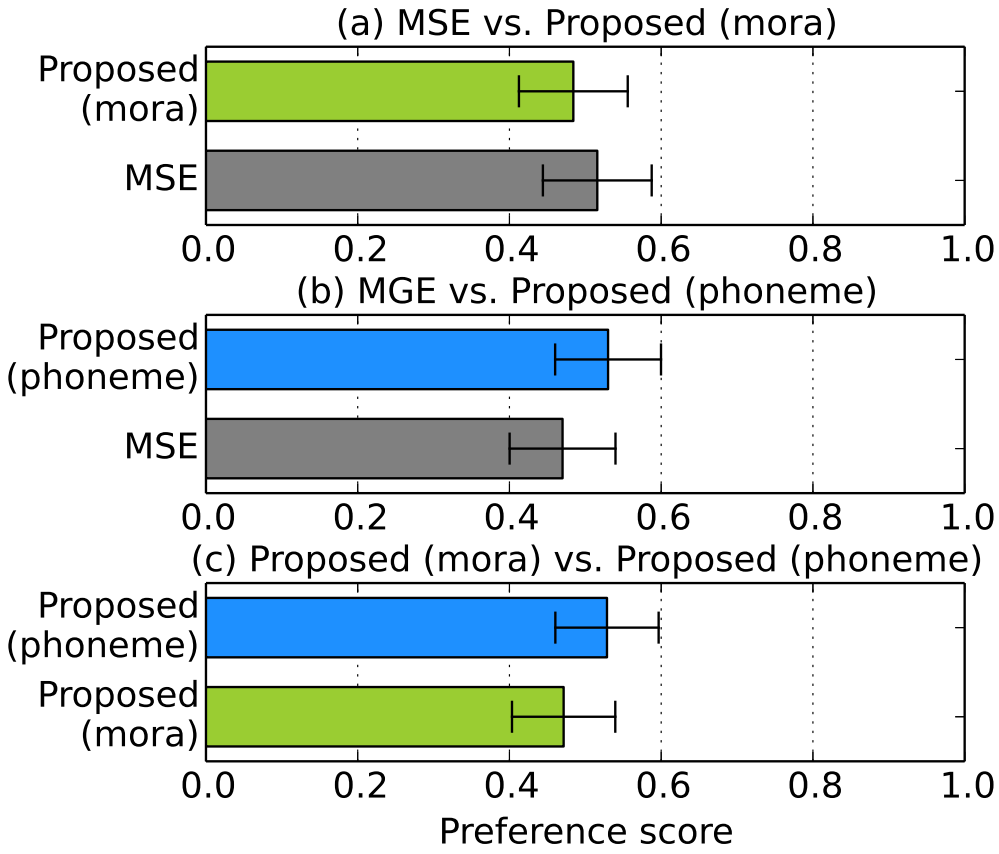


Fig. 3.13: Preference scores of speech quality with 95% confidence intervals (duration generation in TTS). From the top, the numbers of the listeners were 19, 20, and 21, respectively.

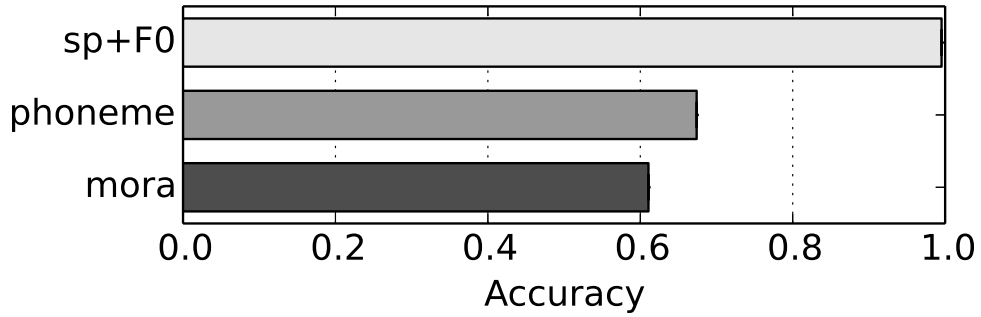


Fig. 3.14: Accuracy of discriminator. "sp+F0", "phoneme", and "mora" denote using the spectral parameters and $F_0$, phoneme durations, and mora durations for discriminating the natural and synthetic speech, respectively.

more gain in speech quality than the conventional GV compensation.

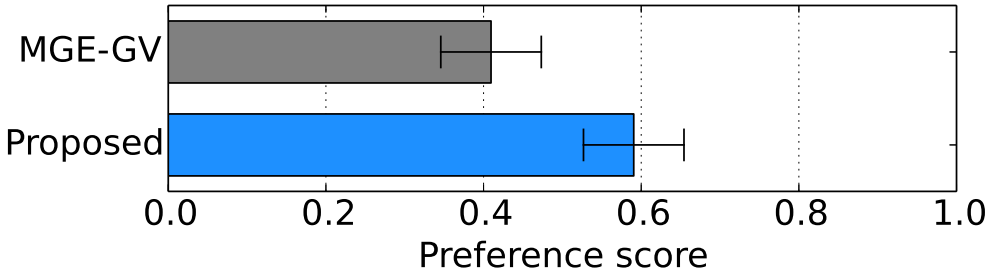


Fig. 3.15: Preference scores of speech quality with 95% confidence intervals (compared to the GV compensation).

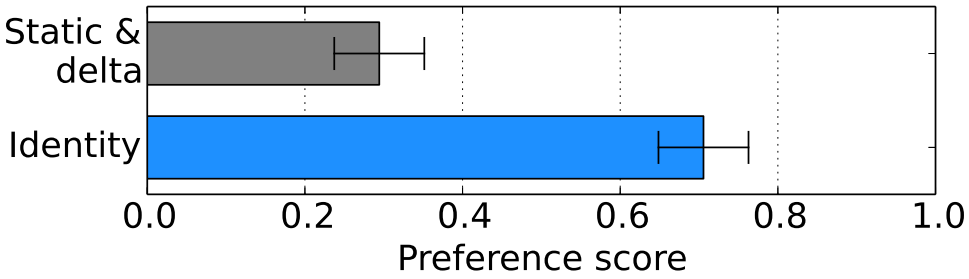


Fig. 3.16: Preference scores of speech quality with 95% confidence intervals (effect of the feature function which is used in anti-spoofing).

### 3.4.8 Effect of Feature Function

We investigate whether the feature function used in anti-spoofing is effective to our method. We adopted the following two functions:

Identity: $\boldsymbol{\phi}(\boldsymbol{y}) = \boldsymbol{y}$
Static & delta [47]: $\boldsymbol{\phi}(\boldsymbol{y}) = \boldsymbol{M}\boldsymbol{y}$

"Identity" is equivalent to not using the feature function. When "Static & delta" is adopted, joint vectors of the static, delta, and delta-delta mel-cepstral coefficients and continuous $F_0$ are input to the discriminator. A preference AB test with 31 listeners was conducted by using our crowd-sourced subjective evaluation systems.

Figure 3.16 shows the results. Clearly, the score of "Static & delta" is much lower than that of "Identity." From this result, although "Static & delta" effectively distinguishes natural and synthetic speech, it does not improve speech quality.

### 3.4.9 Subjective Evaluation Using Complicated DNN Architecture

Only simple Feed-Forward networks were used in the above-described evaluations. Accordingly, we confirm whether our method can improve speech quality even when more

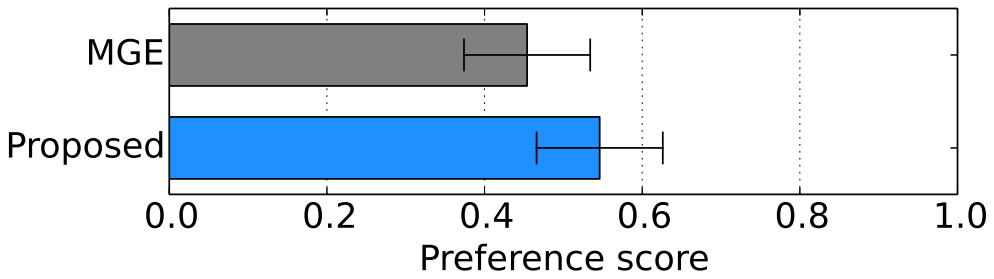


Fig. 3.17: Preference scores of speech quality with 95% confidence intervals (comparison in using LSTMs).

complicated networks are used. We used two-layer uni-directional LSTMs [62] as both acoustic models and discriminator. The numbers of memory cells in the acoustic models and discriminator were 256 and 128, respectively. Our method was applied to spectral and $F_0$ parameters. MGE ("MGE") and the proposed ("Proposed") training algorithms were compared. A preference AB test with 19 listeners was conducted by using our crowd-sourced subjective evaluation systems.

Figure 3.17 shows the results. Since the score of "Proposed" is higher than that of "MGE," we can demonstrate that our method works for not only simple architectures but also complicated ones.

### 3.4.10 Effect of Divergence to Be Minimized by GANs

As the final investigation regarding TTS, we compared speech qualities of various GANs. We adopted the following GANs:

GAN: Eqs. (3.1) and (3.2)
KL-GAN: Eqs. (3.6) and (3.7)
RKL-GAN: Eqs. (3.9) and (3.10)
JS-GAN: Eqs. (3.12) and (3.13)
W-GAN: Eqs. (3.15) and (3.16)
LS-GAN: Eqs. (3.17) and (3.18)

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 3.18 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,

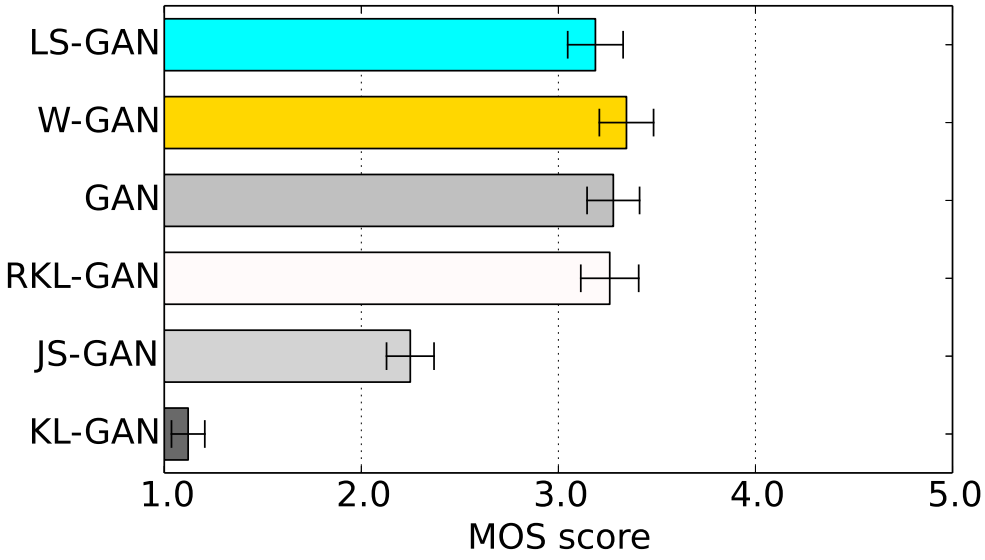


Fig. 3.18: MOS scores of speech quality with 95% confidence intervals (comparison in divergences of GANs).

and KL-GAN.

## 3.5 Experimental Evaluations for VC

### 3.5.1 Conditions for VC Evaluation

The experimental conditions such as the 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. The effectiveness of the proposed algorithm was investigated in 1) VC using speech parameter conversion, and 2) VC using spectral differentials.

In evaluation using speech parameter conversion, DNNs for male-to-male conversion and male-to-female conversion were constructed. Feed-Forward DNNs were adopted to the acoustc models and discriminator. 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. The DTW algorithm was used to align total frame lengths of the input and output speech parameters.

In evaluation using spectral differentials, DNNs for male-to-male conversion were constructed. Here, instead of Feed-Forward DNNs, input-to-output highway networks [79] (described in Appendix A) were adopted to acoustic models. The transform gate of the

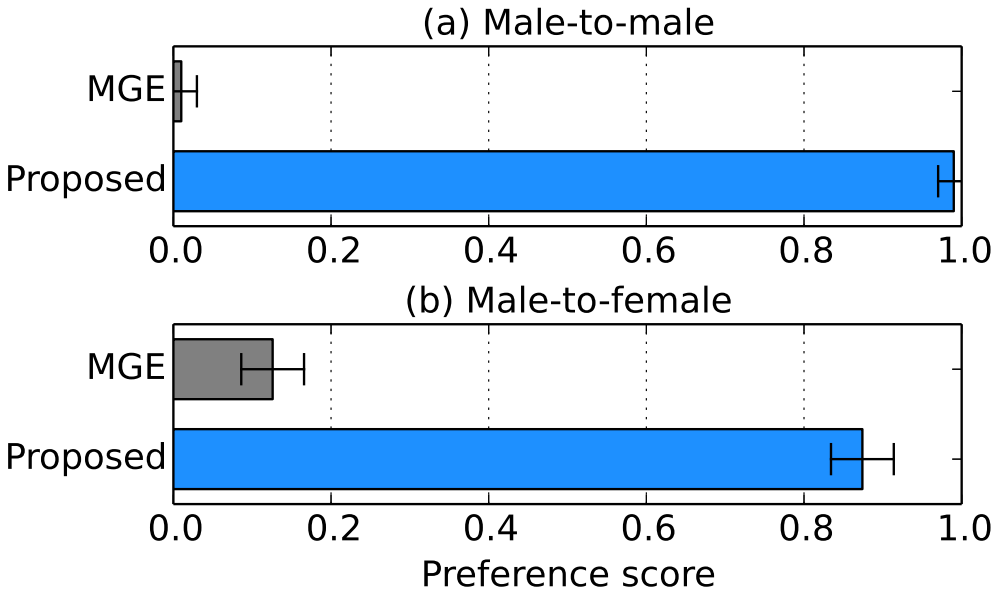


Fig. 3.19: Preference scores of speech quality with 95% confidence intervals (DNN-based VC using speech parameter conversion).

highway networks only had a 59-unit input and 59-unit sigmoid output layers.

We generated speech samples with the conventional MGE training and the proposed training algorithms. We conducted preference AB tests 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, XAB tests on the speaker individuality were conducted using the natural speech as a reference "X."

### 3.5.2 Subjective Evaluation Using Speech Parameter Conversion

In the subjective evaluations, 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 Figs. 3.19 and 3.20, respectively. We can find that our algorithm achieves better scores in speech quality the same as 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 [11]. These improvements were observed not only in the inter-gender but also cross-gender cases.

### 3.5.3 Subjective Evaluation Using Spectral Differentials

Eight listeners participated in the evaluations. The results of the preference tests on speech quality and speaker individuality are shown in Fig. 3.21 and 3.22, respectively.

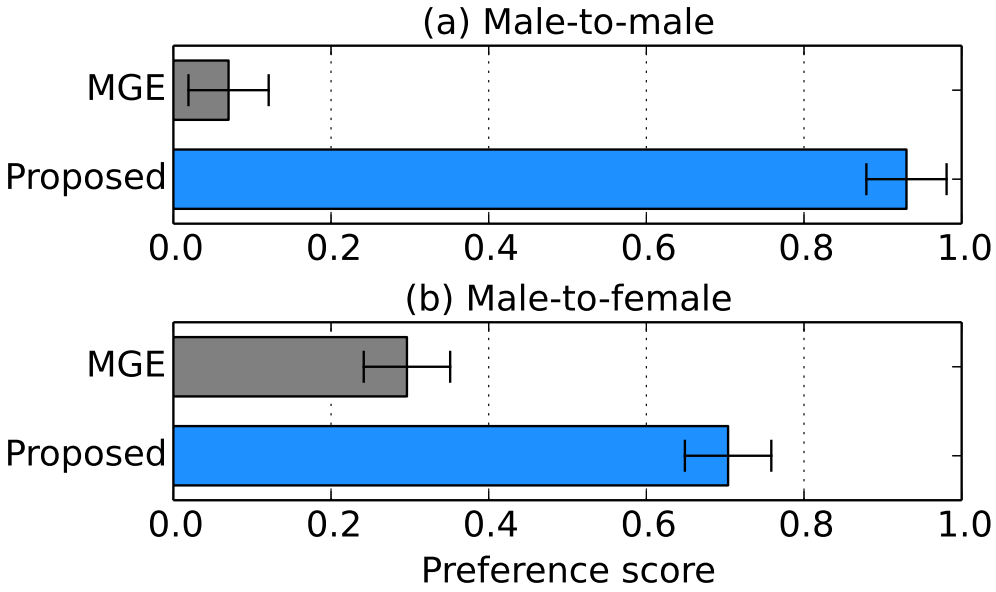


Fig. 3.20: Preference scores of speaker individuality with 95% confidence intervals (DNN-based VC using speech parameter conversion).

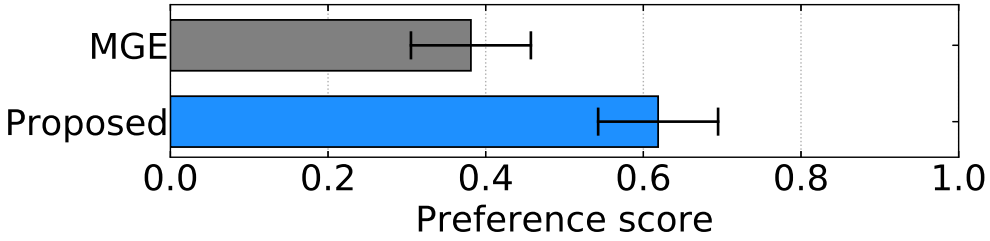


Fig. 3.21: Preference scores of speech quality with 95% confidence intervals (DNN-based VC using spectral differentials).

The results demonstrate that the proposed algorithm also effective in VC using spectral differentials, although the improvements of the scores decreased compared to those shown in Figs. 3.19 and 3.20.

## 3.6 Summary

This chapter proposed a novel training algorithm for DNN-based high-quality SPSS. The algorithm incorporates a framework of GANs, which adversarily trains 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

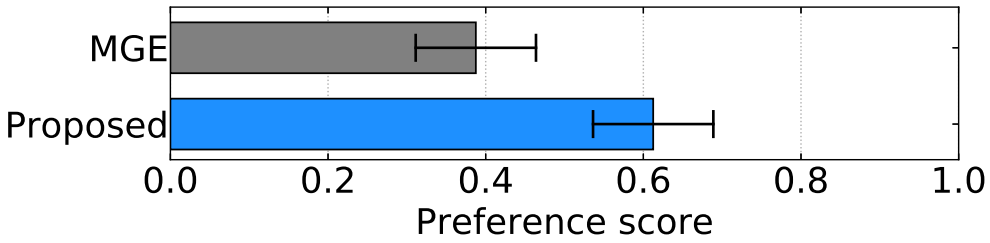


Fig. 3.22: Preference scores of speaker individuality with 95% confidence intervals (DNN-based VC using spectral differentials).

that of natural speech. It was found that the proposed algorithm compensated not only global variance but also correlation among generated speech parameters. Experimental evaluations were conducted in both DNN-based TTS and VC. The results demonstrated that the proposed algorithm yielded significant improvements in terms of speech quality in both TTS and VC regardless of its hyper-parameter settings. The results also showed that the proposed algorithm incorporating the Wasserstein GAN improved synthetic speech quality the most in comparison with various GANs.

# Chapter 4

# Vocoder-free Statistical Parametric Speech Synthesis Using GANs

## 4.1 Introduction

This chapter extends the algorithm proposed in Chapter 3 to vocoder-free SPSS using STFT spectra. In modeing STFT spectra, we must deal with the difficulty due to the high dimensionality of the features. Moreover, since the STFT spectra include both of the spectral parameters and excitation parameters as described in Section 2.2.1, their distribution is more complicated than that of the conventional vocoder-derived speech parameters. To overcome these difficulties, this chapter proposes low- and multi-resolution GAN-based training algorithms. In the proposed algorithm that uses the low-resolution GANs, acoustic models are trained to minimize the weighted sum of the mean squared error between natural and generated spectra in the original resolution and adversarial loss to deceive a discriminator in the lower resolution. Since the low-resolution spectra are close to filter banks and their distribution becomes simpler, we can expect that the GAN-based distribution compensation works well. Furthermore, this chapter proposes an algorithm using multi-resolution GANs, which uses both the low-resolution GANs and original-resolution GANs.

This chapter is organized as follows. Section 4.2 describes the proposed algorithms using low- and multi-resolution GANs. Section 4.3 presents experimental evaluations of the proposed algorithms in DNN-based TTS using STFT spectra. Section 4.4 summarizes this chapter.

## 4.2 Acoustic Model Training Using Low-/Multi-resolution GANs

### 4.2.1 Acoustic Model Training Criteria Using Low-resolution GANs

The algorithm described in Section 3.2 can be applied to STFT spectra generation. However, it suffers from a higher dimensionality and complex distribution of the spectral

amplitudes. We introduce a low-resolution discriminator $D^{(\mathrm{L})}(\cdot)$, which distinguishes natural and generated STFT spectra in the low-frequency resolution. Let $\boldsymbol{\phi}(\cdot)$ be an average-pooling function that converts the spectral amplitudes in the original-frequency resolution $\boldsymbol{y}$ into those in the low-frequency resolution, $\boldsymbol{y}^{(\mathrm{L})} = [\boldsymbol{y}_1^\top, \cdots, \boldsymbol{y}_t^\top, \cdots, \boldsymbol{y}_T^\top]^\top$. The $d$-th frequency bin of the low-resolution spectra at frame $t$, $y_t^{(\mathrm{L})}(d)$, is calculated as

$$y_t^{(\mathrm{L})}(d) = \frac{1}{w} \sum_{i=-p+(d-1)s}^{-p+(d-1)s+w} y_t(i), \tag{4.1}$$

where $p$, $w$, and $s$ denote the size of zero-padding, width of pooling window, and stride of pooling, respectively. The term $y_t(i)$ takes 0 if $i < 0$ or $i > D_y$. Here, $D_y$ is equal to the total number of frequency bins in the original-frequency resolution. The total number of frequency bins in the low-frequency resolution $D_y^{(\mathrm{L})}$ is given as

$$D_y^{(\mathrm{L})} = \frac{D_y + 2p - w}{s} + 1. \tag{4.2}$$

The above processes are similar to conversion from a raw STFT spectra into the filter-bank parameters that represent spectral envelopes of speech. The loss function for training the acoustic models is defined as follows:

$$L_{\mathrm{G}}^{(\mathrm{Low})}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = L_{\mathrm{MSE}}(\boldsymbol{y}, \hat{\boldsymbol{y}}) + \omega_{\mathrm{D}}^{(\mathrm{L})} \frac{\mathbb{E}_{\hat{\boldsymbol{y}}}[L_{\mathrm{MSE}}]}{\mathbb{E}_{\hat{\boldsymbol{y}}^{(\mathrm{L})}}[L_{\mathrm{ADV}}]} L_{\mathrm{ADV}}^{(\mathrm{GAN})}\left(\hat{\boldsymbol{y}}^{(\mathrm{L})}\right), \tag{4.3}$$

where $\hat{\boldsymbol{y}}^{(\mathrm{L})} = \boldsymbol{\phi}(\hat{\boldsymbol{y}})$, and $\omega_{\mathrm{D}}^{(\mathrm{L})}$ is a hyperparameter to control the effect of the second term. This loss function can be regarded as the weighted sum of the MSE in the original resolution and adversarial loss in the lower resolution. Since the distributions of $\boldsymbol{y}^{(\mathrm{L})}$ and $\hat{\boldsymbol{y}}^{(\mathrm{L})}$ are simpler than those of $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$, we can overcome the difficulties in the training due to the high dimensionality and complex distribution. Also, we can expect the low-resolution GAN to dramatically improve the synthetic speech quality because it can capture the difference between spectral envelopes of natural and synthetic speech, which are dominant features in terms of speech quality. The low-resolution discriminator are trained in the same manner as in Eq. (3.1), but $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$ are replaced with $\boldsymbol{y}^{(\mathrm{L})}$ and $\hat{\boldsymbol{y}}^{(\mathrm{L})}$, respectively.

### 4.2.2 Acoustic Model Training Criteria Using Multi-resolution GANs

The proposed algorithm that uses the low-resolution GAN described in Section 4.2.1 can be extended to use multi-resolution GANs, which introduces not only the low-resolution discriminator $D^{(\mathrm{L})}(\cdot)$ but also original-resolution discriminator $D(\cdot)$. The loss function for training the acoustic models is defined as follows:

$$\begin{aligned} L_{\mathrm{G}}^{(\mathrm{Multi})}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = L_{\mathrm{MSE}}(\boldsymbol{y}, \hat{\boldsymbol{y}}) &+ \omega_{\mathrm{D}} \frac{\mathbb{E}_{\hat{\boldsymbol{y}}}[L_{\mathrm{MSE}}]}{\mathbb{E}_{\hat{\boldsymbol{y}}}[L_{\mathrm{ADV}}]} L_{\mathrm{ADV}}^{(\mathrm{GAN})}(\hat{\boldsymbol{y}}) \\ &+ \omega_{\mathrm{D}}^{(\mathrm{L})} \frac{\mathbb{E}_{\hat{\boldsymbol{y}}}[L_{\mathrm{MSE}}]}{\mathbb{E}_{\hat{\boldsymbol{y}}^{(\mathrm{L})}}[L_{\mathrm{ADV}}]} L_{\mathrm{ADV}}^{(\mathrm{GAN})}\left(\hat{\boldsymbol{y}}^{(\mathrm{L})}\right). \end{aligned} \tag{4.4}$$

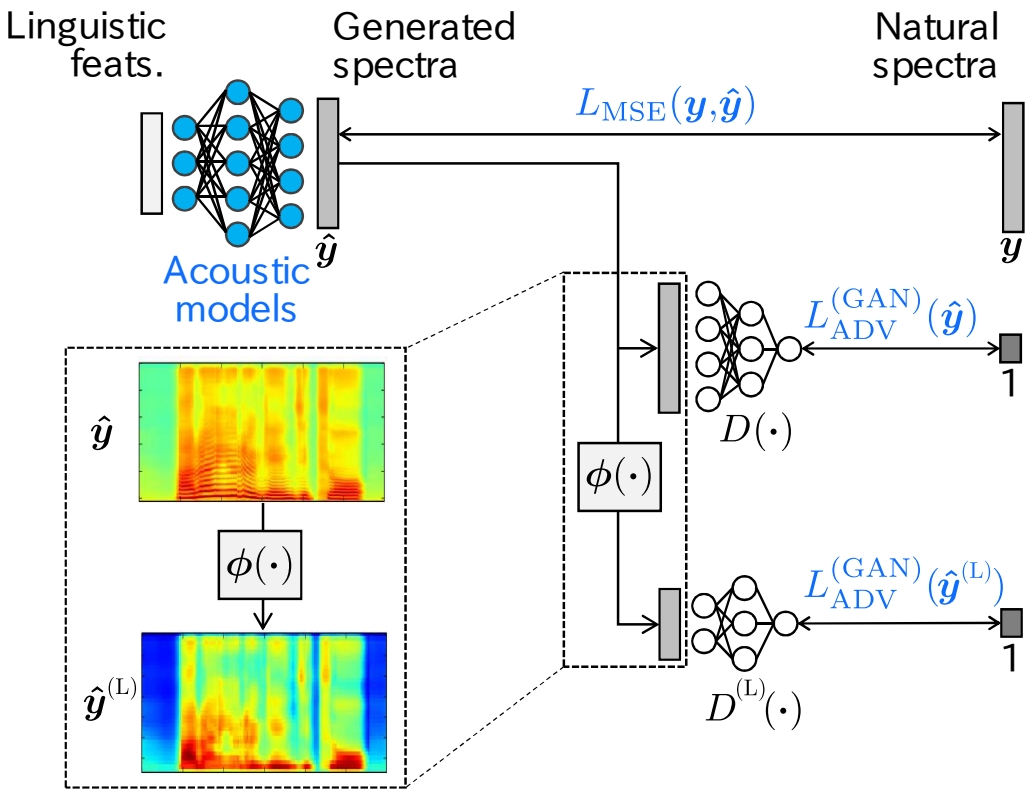


Fig. 4.1: Loss functions for updating acoustic models in proposed algorithm using multi-resolution GANs. $\phi(\cdot)$ is an average-pooling function to convert STFT spectral amplitudes into low-resolution spectra.

When $\omega_{\mathrm{D}} = 0$, this loss function is the same as that in Eq. (4.3). Figure 4.1 illustrates the computation procedure of the loss function. Note that the discriminators are trained separately.

### 4.2.3 Discussions

Kaneko et al. [80] proposed a GAN-based post-filter for STFT spectra. As explained in Section 3.3.5, this post-filter-based approach requires additional computation in synthesis, but our algorithms do not. Also, because the previous work splits the STFT spectra into several sub-frequency bands and applies GANs to each band independently, it ignores the overall spectral structures (i.e., spectral envelope) and their correlation. On the other hand, our algorithms can effectively capture them.

The average pooling function used in the proposed algorithm with the low-resolution GANs can be regarded as the feature function described in Section 3.3.2. By changing the setting of the pooling, we can also compensate for the difference between natural and synthetic speech in the domain of mel-filter banks [47].

By shifting our proposed method from SPSS using vocoders (described in Chapter 3) to vocoder-free SPSS using STFT spectra (described in this chapter), we expect that it

will become easier to extend the algorithm, e.g., waveform-level GANs, in the future.

## 4.3 Experimental Evaluations

### 4.3.1 Experimental Conditions

We used speech data of a Japanese female speaker who uttered 4007 sentences (some of the JSUT corpus [81]). The number of utterances included in training and evaluation data were 3808 and 199, respectively. Speech signals were sampled at a rate of 16 kHz. The frame length, shift length, and FFT length were set to 400 (25 ms), 80 (5 ms), and 1024 samples, respectively. We used the hamming window for FFT analysis. In the training phase, linguistic features which have a real value, and log spectral amplitudes were normalized to have zero-mean unit-variance. We removed 90% of the silence frames from the training data to improve training accuracy.

The DNN architectures for acoustic models and discriminator were Feed-Forward. The input of the acoustic models were 444-dimensional vectors including 439-dimensional linguistic features, 3-dimensional duration features, continuous log $F_0$, and U/V. The $F_0$ was extracted from speech data by using STRAIGHT vocoder systems [24]. We constructed DNNs, which predicted duration and $F_0$ features from linguistic features, in advance. The architecture for the acoustic models included $3 \times 1024$-unit ReLU [35] hidden layers and a 513-unit linear output layer. The architecture for the discriminator in the original resolution included $3 \times 512$-unit ReLU hidden layers and one unit sigmoid output layer. The architectures for the discriminator in the lower resolution were almost same as that in the original resolution; that is, the activation functions used in the hidden and output layers were ReLU and sigmoid, the number of hidden layers was 3, but the number of input and hidden units varied in accordance with the parameters of the pooling function $\boldsymbol{\phi}(\cdot)$. In the following experiments, we fixed $p = 6$ and $s = w/2$ in Eq. (4.2). $w$ was set to 14, 30, and 70. Accordingly, the number of input units $D_y^{(\mathrm{L})}$ was set to 74, 34, 14, and the number of hidden units was set to 128, 64, 32, respectively.

In the training phase, we initialized the acoustic models by minimizing the MSE (described in Section 2.4.2) with 25 iterations. "Iteration" means using all the training data (3808 utterances) once for training. The discriminators in the original and lower resolution were initialized using natural speech and generated spectra after the initialization of the acoustic models. The number of iterations for the initialization was 5. The proposed training algorithms were used with 25 iterations. The expectation values for scaling the loss functions were estimated at each iteration step. We used AdaGrad [82] as the optimization algorithm, setting the learning rate to 0.01.

We conducted subjective evaluations on the quality of the synthetic speech with various hyperparameter settings. Preference AB tests were conducted to evaluate the quality of speech produced from several algorithms. 25 listeners participated in each of the following evaluations by using our crowd-sourced evaluation systems, and each listener evaluated 10

Table 4.1: Preference scores of speech quality with their $p$-values (original-resolution GANs)

| | Score | $p$-value | |
|---|---|---|---|
| Baseline | **0.700** vs. 0.300 | $< 10^{-10}$ | $\omega_{\mathrm{D}} = 0.5$ |
| $\omega_{\mathrm{D}} = 1.0$ | 0.280 vs. **0.720** | $< 10^{-10}$ | Baseline |
| $\omega_{\mathrm{D}} = 0.5$ | 0.496 vs. **0.504** | $8.6 \times 10^{-1}$ | $\omega_{\mathrm{D}} = 1.0$ |

samples. The total number of listeners was 375. In the following evaluations, "Baseline" denotes the method that trains the acoustic models using conventional MSE loss [18], i.e., both hyperparameters, $\omega_{\mathrm{D}}$ and $\omega_{\mathrm{D}}^{(\mathrm{L})}$ in Eq. (4.4), were set to 0.

### 4.3.2 Subjective Evaluation of Original-resolution GANs

First, to investigate the effect of GAN-based training in the original resolution (i.e., the same as the algorithm described in the previous chapter), we fixed $\omega_{\mathrm{D}}^{(\mathrm{L})} = 0$, and set $\omega_{\mathrm{D}} = 0.5$ or 1.0. We compared the quality of "Baseline" and our proposed algorithm using original-resolution GANs with "$\omega_{\mathrm{D}} = 0.5$," and "$\omega_{\mathrm{D}} = 1.0$." Table 4.1 shows the experimental results. Compared with "Baseline," the methods using the original-resolution GANs significantly degraded synthetic speech quality regardless of the hyperparameter settings. Therefore, we can confirm that simply applying the GAN-based training algorithm, which is effective in conventional SPSS using vocoders [83], does not improve STFT spectra generation.

### 4.3.3 Subjective Evaluation of Low-resolution GANs

Next, to investigate effect of $w$, we fixed $\omega_{\mathrm{D}} = 0$ and set $\omega_{\mathrm{D}}^{(\mathrm{L})} = 1$. We compared the quality of generated speech samples using "Baseline" and our algorithm using the low-resolution GANs win "$w = 14$," "$w = 30$," and "$w = 70$." Table 4.2 shows the experimental results. From the results shown in Table 4.2(a), we can see that the proposed algorithm using the low-resolution GANs always achieved better scores than "Baseline," regardless of its parameter settings of the pooling function, which demonstrates the effectiveness of this algorithm. We set $w$ to 30 in the following evaluation because Table 4.2(b) shows that "$w = 30$" was the best, although there were no significant differences among the scores.

We also investigated the effect of the hyperparameter in the low-resolution GANs. We fixed $\omega_{\mathrm{D}} = 0$ and set $\omega_{\mathrm{D}}^{(\mathrm{L})} = 0.5$ or 1.0. We compared the quality of generated speech using "Baseline" and our algorithm using the low-resolution GANs with "$\omega_{\mathrm{D}}^{(\mathrm{L})} = 0.5$," and "$\omega_{\mathrm{D}}^{(\mathrm{L})} = 1.0$." Table 4.3 shows the experimental results. From the results, we can conclude that the proposed algorithm using the low-resolution GANs successfully improved synthetic speech quality regardless of its hyperparameter settings.

Table 4.2: Preference scores of speech quality with their $p$-values (low-resolution GANs with various pooling-parameter settings)

(a) Results of comparing "Baseline" with using low-resolution GANs

| | Score | $p$-value | |
|---|---|---|---|
| $w = 14$ | **0.568** vs. 0.432 | $2.3 \times 10^{-3}$ | Baseline |
| $w = 30$ | **0.572** vs. 0.428 | $1.2 \times 10^{-3}$ | Baseline |
| $w = 70$ | **0.528** vs. 0.472 | $2.1 \times 10^{-1}$ | Baseline |

(b) Results of proposed algorithm using low-resolution GANs

| | Score | $p$-value | |
|---|---|---|---|
| $w = 14$ | 0.488 vs. **0.512** | $5.9 \times 10^{-1}$ | $w = 30$ |
| $w = 30$ | **0.532** vs. 0.468 | $1.5 \times 10^{-1}$ | $w = 70$ |
| $w = 70$ | 0.472 vs. **0.528** | $2.1 \times 10^{-1}$ | $w = 14$ |

Table 4.3: Preference scores of speech quality with their $p$-values (low-resolution GANs with various hyperparameter settings)

| | Score | $p$-value | |
|---|---|---|---|
| Baseline | 0.456 vs. **0.544** | $4.9 \times 10^{-2}$ | $\omega_{\mathrm{D}}^{(\mathrm{L})} = 0.5$ |
| $\omega_{\mathrm{D}}^{(\mathrm{L})} = 1.0$ | **0.588** vs. 0.412 | $7.6 \times 10^{-5}$ | Baseline |
| $\omega_{\mathrm{D}}^{(\mathrm{L})} = 0.5$ | **0.504** vs. 0.496 | $8.6 \times 10^{-1}$ | $\omega_{\mathrm{D}}^{(\mathrm{L})} = 1.0$ |

### 4.3.4 Subjective Evaluation of Multi-resolution GANs

Finally, we examined the effects of the proposed algorithm using the multi-resolution GANs. We generated speech samples using the following algorithms:

Original: $(\omega_{\mathrm{D}}, \omega_{\mathrm{D}}^{(\mathrm{L})}) = (1.0, 0.0)$
Low: $(\omega_{\mathrm{D}}, \omega_{\mathrm{D}}^{(\mathrm{L})}) = (0.0, 1.0)$
Multi: $(\omega_{\mathrm{D}}, \omega_{\mathrm{D}}^{(\mathrm{L})}) = (1.0, 1.0)$

Table 4.4 shows the results, Obviously, the proposed algorithm using the low-resolution GANs achieved a much higher score than the others. To investigate this reason, we plotted the STFT spectral magnitudes of synthetic speech used for the evaluations illustrated in Fig. 4.2. We can see that high randomness observed in natural spectra (Fig. 4.2(a)) was excessively smoothed in synthetic speech of "Baseline" (Fig. 4.2(b)), while the proposed three algorithms reproduced the randomness by using GANs, However, there were some temporal discontinuities in the spectra generated by using original- and multi-resolution

Table 4.4: Preference scores of speech quality with their $p$-values (multi-resolution GANs)

| | Score | $p$-value | |
|---|---|---|---|
| Low | **0.808** vs. 0.192 | $< 10^{-10}$ | Multi |
| Multi | 0.492 vs. **0.508** | $7.2 \times 10^{-1}$ | Original |
| Original | 0.192 vs. **0.808** | $< 10^{-10}$ | Low |

GANs (Figs. 4.2(d) and (e)), which might considerably degrade the synthetic speech quality. One can address the quality degradation by using recurrent architectures such as long-short term memory [33, 62] for the acoustic models and discriminator to make them capture the temporal dependency of the STFT spectra. Further improvements also can be achieved by conditioning the GANs with the specific information of the utterance such as the phonetic contents, and U/V [84].

## 4.4 Summary

This chapter proposed two training algorithms to incorporate GANs into vocoder-free SPSS using STFT spectra. In the proposed algorithm using a low-resolution GANs, acoustic models are trained to minimize the MSE between natural and generated STFT spectral amplitudes at the original resolution and the distribution differences of their distributions at low resolution. This algorithm can be extended to one using multi-resolution GANs, which also minimizes the distribution differences of natural and generated STFT spectra at the original resolution. Experimental results indicated that the algorithm using the original-resolution GANs and our proposed algorithm using multi-resolution GANs degraded synthetic speech quality, but the proposed algorithm using the low-resolution GANs successfully improved it.

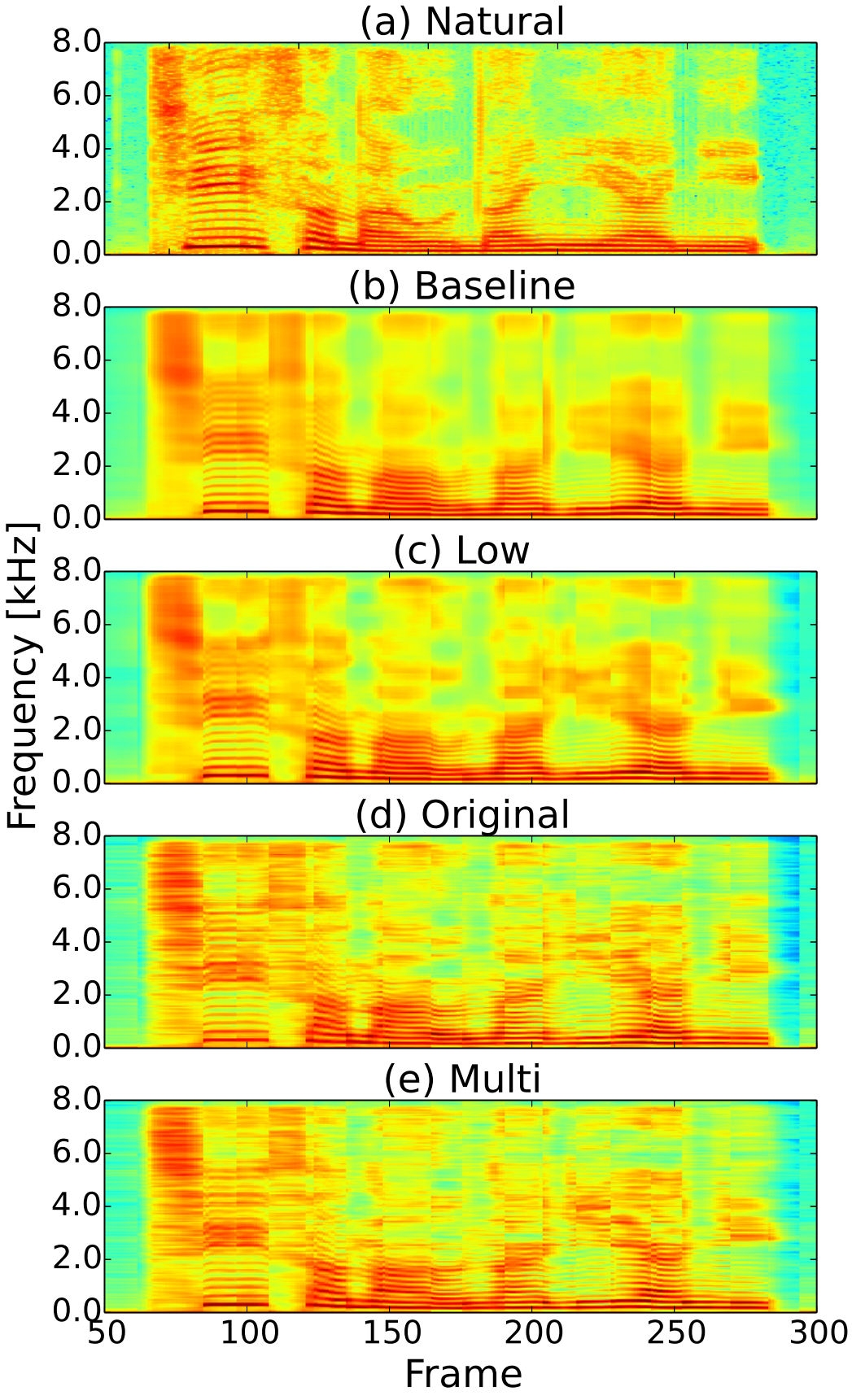


Fig. 4.2: STFT spectral magnitudes of natural and synthetic speech.

# Chapter 5

# Conclusion

## 5.1 Thesis Summary

The advantage of SPSS is the flexibility to control the characteristics of the synthetic speech. However, several factors degrade speech quality. The primary one is over-smoothing of the generated speech parameters due to acoustic modeling. Another is vocoder-derived speech parameterization, which can be avoided by using vocoder-free SPSS, such as TTS using STFT spectra. This thesis addressed these factors by introducing GANs into the acoustic model training of SPSS.

In Chapter 2, we reviewed conventional SPSS using DNNs. First, the basic framework of SPSS using vocoder-derived speech parameters was described. GV compensation, a conventional method for alleviating the over-smoothing effect, was also described. A method that goes beyond SPSS using vocoder systems was also described: TTS using STFT spectra. Feed-Forward DNN and LSTM, which are often used as acoustic models, were described. The loss functions mainly used for acoustic model training were explained.

In Chapter 3, a novel training algorithm incorporating GANs was proposed. Acoustic models are trained to minimize the weighted sum of the conventional MGE loss and adversarial loss, which makes the discriminator recognize generated speech parameters as natural. Because this GAN-like training criterion reduces the distribution difference between natural and generated speech parameters, the proposed algorithm can reproduce not only GVs but also correlations among generated speech parameters. Moreover, the discriminator in the proposed algorithm can be regarded as anti-spoofing using DNNs. Thus, techniques for anti-spoofing can be incorporated into the proposed algorithm. Experimental evaluations of the proposed algorithm were conducted for DNN-based TTS and VC. The TTS evaluation showed that the proposed algorithm improved synthetic speech quality regardless of the hyperparameter settings used to control the weights for the adversarial loss. The results of an investigation focused on the divergences minimized by the GAN demonstrated that the W-GAN, which minimizes the Earth Mover's distance, was the best among various GANs at improving the quality of the synthetic speech. The VC evaluation showed that the proposed algorithm outperformed the conventional MGE

training algorithm in terms of both converted speech quality and speaker individuality.

In Chapter 4, the algorithm described in Chapter 3 was extended to vocoder-free SPSS using STFT spectra. To address the difficulty in acoustic model training due to the complex distribution of STFT spectral amplitudes, training algorithms using low- and multi-resolution GANs were presented. Use of the one using low-resolution GANs reduced the difference in spectral envelopes of the natural and generated STFT spectra. This can be extended to the algorithm using multi-resolution GANs, which uses low- and original-resolution discriminators for training acoustic models. Experimental results demonstrated that the algorithm using low-resolution GANs improved synthetic speech quality and worked robustly against its hyperparameter settings. Comparison of the original-, low-, and multi-resolution GANs revealed that the low-resolution GANs were the best for improving synthetic speech quality.

## 5.2 Future Work

Although we have improved synthetic speech quality of SPSS, several problems remain to be solved.

### 5.2.1 Investigating or Devising Effective Techniques for Anti-spoofing

As described in Section 3.3.2, anti-spoofing techniques can be integrated into the proposed algorithm. However, the use of the dynamic features of the spectral parameters, which is effective for detecting voice spoofing attacks, did not improve synthetic speech quality, as shown in Fig. 3.16. One possible solution is to use anti-spoofing techniques related to human perception in speech.

### 5.2.2 Further Improving Synthetic Speech Quality using STFT Spectra

The proposed algorithm using low-resolution GANs improved synthetic speech quality. Further improvements can be made by developing methods for effectively using the original-resolution discriminator. Introduction of conditional GANs into the proposed algorithm should make the discriminator capture more of the fine structures of natural STFT spectral amplitudes.

# Publications and Research Activities

**Original Journal Papers**

1. **Yuki Saito**, Shinnosuke Takamichi, and Hiroshi 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, Jan. 2018. (**The 33rd TELECOM System Technology Award for Students from TAF**)
2. **Yuki Saito**, Shinnosuke Takamichi, and Hiroshi Saruwatari, "Voice conversion using input-to-output highway networks," *IEICE Transactions on Information and Systems,* vol. E100-D, no. 8, pp. 1925–1928, Aug. 2017.

**International Conferences (Peer-Reviewed)**

1. **Yuki Saito**, Yusuke Ijima, Kyosuke Nishida, and Shinnosuke Takamichi, "Non-parallel voice conversion using variational autoencoders conditioned by phonetic posteriorgrams and d-vectors," *Proc. ICASSP,* Alberta, Canada, Apr. 2018. (accepted)
2. **Yuki Saito**, Shinnosuke Takamichi, and Hiroshi Saruwatari, "Text-to-speech synthesis using STFT spectra based on low-/multi-resolution generative adversarial networks," *Proc. ICASSP,* Alberta, Canada, Apr. 2018. (accepted)
3. Hiroyuki Miyoshi, **Yuki Saito**, Shinnosuke Takamichi, and Hiroshi Saruwatari, "Voice conversion using sequence-to-sequence learning of context posterior probabilities," *Proc. INTERSPEECH,* pp. 1268–1272, Stockholm, Sweden, Aug. 2017.
4. **Yuki Saito**, Shinnosuke Takamichi, and Hiroshi Saruwatari, "Training algorithm to deceive anti-spoofing verification for DNN-based speech synthesis," *Proc. ICASSP,* pp. 4900–4904, New Orleans, U.S.A., Mar. 2017. (**Spoken Language Processing Student Grant of ICASSP 2017**)
5. **Yuki Saito**, and Hiroshi Tenmoto, "Construction of highly interpretable classification rule based on linear SVM," Proc. ISTS, Taipei, Taiwan, Nov. 2014.

### Technical Reports

1. **Yuki Saito**, Yusuke Ijima, Kyosuke Nishida, and Shinnosuke Takamichi, "Non-parallel and many-to-many voice conversion using variational autoencoders using phonetic posteriorgrams and d-vectors," *Proc. IEICE Technical Report,* Mar. 2018. (to appear, in Japanese)
2. Masakazu Une, **Yuki Saito**, Shinnosuke Takamichi, Daichi Kitamura, Ryoichi Miyazaki, and Hiroshi Saruwatari, "Generative adversarial training of the noise generation model for speech synthesis using speech in noise," *IPSJ SIG Technical Report,* 2017-SLP-118, no. 1, pp. 1-6, Oct. 2017. (in Japanese)
3. Hiroyuki Miyoshi, **Yuki Saito**, Shinnosuke Takamichi, and Hiroshi Saruwatari, "Voice conversion using sequence-to-sequence learning of context posterior probabilities and evaluation of the dual learning," *IEICE Technical Report,* SP2017-16, vol. 117, No. 160, pp. 9–14, Jul. 2017. (in Japanese)
4. **Yuki Saito**, Shinnosuke Takamichi, and Hiroshi Saruwatari, "Training algorithm to deceive anti-spoofing verification for DNN-based text-to-speech synthesis," *IPSJ SIG Technical Report,* 2017-SLP-115, no. 1, pp. 1–6, Feb. 2017. (in Japanese)
5. **Yuki Saito**, Shinnosuke Takamichi, and Hiroshi Saruwatari, "Evaluation of DNN-based voice conversion deceiving anti-spoofing verification," *IEICE Technical Report,* SP2016-69, vol. 116, no. 414, pp. 29–34, Jan. 2017. (in Japanese, **Student Poster Award**)

### Domestic Conferences

1. Masakazu Une, **Yuki Saito**, Shinnosuke Takamichi, Daichi Kitamura, Ryoichi Miyazaki, and Hiroshi Saruwatari, "Generative approach using the noise generation models for DNN-based speech synthesis trained from noisy speech," *Proc. ASJ, Spring meeting,* Mar. 2018. (to appear, in Japanese)
2. **Yuki Saito**, Shinnosuke Takamichi, and Hiroshi Saruwatari, "Adversarial DNN-based speech synthesis using multi-frequency-resolution STFT spectra," *Proc. ASJ, Spring meeting,* Mar. 2018. (to appear, in Japanese)
3. **Yuki Saito**, Shinnosuke Takamichi, and Hiroshi Saruwatari, "Experimental investigation of divergences in adversarial DNN-based speech synthesis," *Proc. ASJ, Autumn meeting,* 1-8-7, pp. 189–192, Sep. 2017. (in Japanese)
4. Shinnosuke Takamichi, Tomoki Koriyama, **Yuki Saito**, and Hiroshi Saruwatari, "Evaluation of inter-utterance variation in speech synthesis based on moment-matching networks," *Proc. ASJ, Autumn meeting,* 1-8-9, pp. 195–196, Sep. 2017. (in Japanese)
5. **Yuki Saito**, Shinnosuke Takamichi, and Hiroshi Saruwatari, "Adversarial DNN-based voice conversion based on spectral differentials using highway networks," *Proc. ASJ, Spring meeting,* 1-6-14, pp. 235–236, Mar. 2017. (in Japanese, **IEEE Signal Processing Society Tokyo Joint Chapter Student Award**)

6. Hiroyuki Miyoshi, **Yuki Saito**, Shinnosuke Takamichi, and Hiroshi Saruwatari, "Voice conversion using sequence-to-sequence learning of context posterior probabilities," *Proc. ASJ, Spring meeting,* 1-6-15, pp. 237–238, Mar. 2017. (in Japanese)
7. **Yuki Saito**, Shinnosuke Takamichi, and Hiroshi Saruwatari, "F0 contour and duration generation for adversarial DNN-based speech synthesis," *Proc. ASJ, Spring meeting,* 2-6-6, pp. 257–258, Mar. 2017. (in Japanese)
8. **Yuki Saito**, Shinnosuke Takamichi, and Hiroshi Saruwatari, "Training algorithm considering anti-spoofing verification for DNN-based speech synthesis," *Proc. ASJ, Autumn meeting,* 3-5-1, pp. 149–150, Sep. 2016. (in Japanese, **Student Presentation Award of Acoustical Society of Japan**)

**Awards**

1. The 33rd TELECOM System Technology Award for Students from TAF, Mar. 2018.
2. The 1st IEEE Signal Processing Society Tokyo Joint Chapter Student Award, Nov. 2017.
3. Spoken Language Processing Student Grant Award of ICASSP, Mar. 2017.
4. The 14th Best Student Presentation Award of Acoustical Society of Japan, Mar. 2017.
5. 2017 IEICE ISS Student Poster Award, Jan. 2017.
6. Graduation Research Award, Advanced Course of Electronic and Information Systems Engineering, National Institute of Technology, Kushiro College, Feb. 2016.
7. Dean's Award, Department of Information Engineering, National Institute of Technology, Kushiro College, Mar. 2014.

**Misc.**

1. JSPS fellow (DC1, accepted)

# References


[1] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Proc. ICASSP*, Istanbul, Turkey, June 2000, pp. 1315–1318.

[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. 1988.

[4] N. Hattori, T. Toda, H. Kawai, H. Saruwatari, and K. Shikano, "Speaker-adaptive speech synthesis based on Eigenvoice conversion and language-dependent prosodic conversion in speech-to-speech translation," in *Proc. INTERSPEECH*, Florence, Italy, Aug. 2011, pp. 2769–2772.

[5] S. Sitaram, G. Anumanchipalli, J. Chiu, A. Parlikar, and A. W. Black, "Text to speech in new languages without a standardized orthography," in *Proc. SSW8*, pp. 95–100. Barcelona, Spain, Aug. 2013.

[6] H. Doi, T. Toda, T. Nakano, M. Goto, and S. Nakamura, "Singing voice conversion method based on many-to-many Eigenvoice conversion and training data generation using a singing-to-singing synthesis system," in *Proc. APSIPA ASC*, pp. 1–6. Hollywood, U.S.A., Nov. 2012.

[7] K. Kobayashi, T. Toda, T. Nakano, M. Goto, G. Neubig, S. Sakti, and S. Nakamura, "Regression approaches to perceptual age control in singing voice conversion," in *Proc. ICASSP*, Florence, Italy, May 2014, pp. 7954–7958.

[8] H. Zen, K. Tokuda, and A. Black, "Statistical parametric speech synthesis," *Speech Communication*, vol. 51, no. 11, pp. 1039–1064, 2009.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] Y. Ohtani, M. Tamura, M. Morita, T. Kagoshima, and M. Akamine, "Histogram-based spectral equalization for HMM-based speech synthesis using mel-LSP," in *Proc. INTERSPEECH*, Portland, U.S.A., Sep. 2012.

[14] 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.

[15] 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.

[16] K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "Trajectory training considering global variance for speech synthesis based on neural networks," in *Proc. ICASSP*, Shanghai, China, Mar. 2016, pp. 5600–5604.

[17] T. Nose and A. Ito, "Analysis of spectral enhancement using global variance in HMM-based speech synthesis," in *Proc. INTERSPEECH*, Singapore, May 2014, pp. 2917–2921.

[18] S. Takaki, H. Kameoka, and J. Yamagishi, "Direct modeling of frequency spectra and waveform generation based on phase recovery for dnn-based speech synthesis," in *Proc. INTERSPEECH*, Stockholm, Sweden, Aug. 2017, pp. 1128–1132.

[19] A. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," *arXiv*, vol. abs/1609.03499, 2016.

[20] 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*, vol. abs/1612.07837, 2016.

[21] H. Kawahara, Jo Estill, and O. 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 *MAVEBA 2001*, Florence, Italy, Sep. 2001, pp. 1–6.

[22] T. Fukada, K. Tokuda, T. Kobayashi, and S. Imai, "An adaptive algorithm for mel-cepstral analysis of speech," in *Proc. ICASSP*, San Francisco, U.S.A., Mar 1992, pp. 137–140.

[23] K. Yu and S. Young, "Continuous F0 modeling for HMM based statistical parametric speech synthesis," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 5, pp. 1071–1079, 2011.

[24] H. Kawahara, I. Masuda-Katsuse, and A. D. 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. 3–4, pp. 187–207, Apr. 1999.

[25] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: a vocoder-based high-quality speech synthesis system for real-time applications," *IEICE Transactions on Information and Systems*, vol. E99-D, no. 7, pp. 1877–1884, 2016.

[26] M. Morise, "D4C, a band-aperiodicity estimator for high-quality speech synthesis," *Speech Communication*, vol. 84, pp. 57–65, Nov. 2016.

[27] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis," in *Proc. EUROSPEECH*, Budapest, Hungary, Apr. 1999, pp. 2347–2350.

[28] T. Kudo, K. Yamamoto, and Y. Matsumoto, "Applying conditional random fields to Japanese morphological analysis," in *Proc. EMNLP*, Barcelona, Spain, Jul. 2004, pp. 230–237.

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

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

[31] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Proc. ICASSP*, Vancouver, Canada, May 2013, pp. 7962–7966.

[32] 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, 2015.

[33] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[34] S. Fan, Y. Qian, and F. Soong, "TTS synthesis with bidirectional LSTM based recurrent neural networks," in *Proc. INTERSPEECH*, Singapore, Sep. 2014, pp. 1964–1968.

[35] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. AISTATS*, Lauderdale, U.S.A., Apr. 2011, pp. 315–323.

[36] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.

[37] M. C. Mozer, "A focused backpropagation algorithm for temporal pattern recognition," *Complex Systems*, vol. 3, no. 4, pp. 349–381, 1989.

[38] 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.

[39] Z. Wu and S. King, "Improving trajectory modeling for DNN-based speech synthesis by using stacked bottleneck features and minimum trajectory error training,"

*IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 7, pp. 1255–1265, Jul. 2016.

[40] T. Toda and K. Tokuda, "A speech parameter generation algorithm considering global variance for HMM-based speech synthesis," *IEICE Transactions on Information and Systems*, vol. E90-D, no. 5, pp. 816–824, 2007.

[41] T. Toda, T. Muramatsu, and H. Banno, "Implementation of computationally efficient real-time voice conversion," in *Proc. INTERSPEECH*, Portland, U.S.A., Sep. 2012.

[42] S. Imai, K. Sumita, and C. Furuichi, "Mel log spectrum approximation (MLSA) filter for speech synthesis," *Electronics and Communications in Japan*, vol. 66, no. 2, pp. 10–18, 1983.

[43] D. Griffin and J. Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 32, no. 2, pp. 236–243, Apr. 1984.

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

[45] Z. Wu, P. L. D. Leon, C. Demiroglu, A. Khodabakhsh, S. King, Z. Ling, D. Saito, B. Stewart, T. Toda, M. Wester, and J. Yamagishi, "Anti-spoofing for text-independent speaker verification: An initial database, comparison of countermeasures, and human performance," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 4, pp. 768–783, 2016.

[46] N. Chen, Y. Qian, H. Dinkel, B. Chen, and K. Yu, "Robust deep feature for spoofing detection - the SJTU system for ASVspoof 2015 Challenge," in *Proc. INTERSPEECH*, Dresden, Germany, Sep. 2015, pp. 2097–2101.

[47] M. Sahidullah, T. Kinnunen, and C. Hanilçi, "A comparison of features for synthetic speech detection," in *Proc. INTERSPEECH*, Dresden, Germany, Sep. 2015, pp. 2087–2091.

[48] P. L. D. Leon, M. Pucher, J. Yamagishi, I. Hernaez, and I. Saratxaga, "Evaluation of speaker verification security and detection of HMM-based synthetic speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 8, pp. 2280–2290, 2012.

[49] A. Ogihara, H. Unno, and A. Shiozaki, "Discrimination method of synthetic speech using pitch frequency against synthetic speech falsification," *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E88-A, no. 1, pp. 280–286, 2005.

[50] P. L. D. Leon, B. Stewart, and J. Yamagishi, "Synthetic speech discrimination using pitch pattern statistics derived from image analysis," in *Proc. INTERSPEECH*, pp. 370–373. Portland, U.S.A., Sep. 2012.

[51] Z. Wu, X. Xiao, E. S. Chng, and H. Li, "Synthetic speech detection using temporal modulation feature," in *Proc. ICASSP*, Vancouver, Canada, May. 2013, pp. 7234–7238.

[52] G. Esther and E. L. Low, "Durational variability in speech and the rhythm class hypothesis," *Papers in laboratory phonology 7*, pp. 515–546, 2002.

[53] S. Nowozin, B. Cseke, and R. Tomioka, "f-GAN: Training generative neural samplers using variational divergence minimization," in *Proc. NIPS*, Dec. 2016, pp. 271–279.

[54] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," *arXiv*, vol. abs/1701.07875, 2017.

[55] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," *arXiv*, vol. abs/1611.04076, 2017.

[56] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factrization," in *Proc. NIPS*, Colorado, U.S.A., 2000, pp. 556–562.

[57] R. Kompass, "A generalized divergence measure for nonnegative matrix facctorization," *Neural Computation*, vol. 19, no. 3, pp. 780–891, Mar. 2007.

[58] I. Csiszár and P. C. Shields, "Information theory and statistics: A tutorial," *Foundation and Trends in Communications and Information Theory*, vol. 1, no. 4, pp. 417–518, 2004.

[59] Cédric Vilani, *Optimal Transport: Old and New*, Springer, 2009.

[60] B. Huang, D. Ke, H. Zheng, B. Xu, Y. Xu, and K. Su, "Multi-task learning deep neural networks for speech feature denoising," in *Proc. INTERSPEECH*, Dresden, Germany, Sep. 2015, pp. 2464–2468.

[61] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text-to-image synthesis," in *Proc. ICML*, New York, U.S.A., 2016, pp. 1060–1069.

[62] H. Zen and H. Sak, "Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis," in *Proc. ICASSP*, Brisbane, Australia, Apr. 2015, pp. 4470–4474.

[63] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[64] T. R. Marco, S. Sameer, and G. Carlos, ""Why should I trust you?": Explaining the predictions of any classifier," in *Proc. KDD*, San Francisco, U.S.A., Aug. 2016, pp. 1135–1164.

[65] Y. Li, S. Kevin, and Z. Richard, "Generative moment matching networks," in *Proc. ICML*, Lille, France, 2015.

[66] I. Goodfellow, "NIPS 2016 tutorial: Generative adversarial networks," *arXiv*, vol. abs/1701.00160, 2017.

[67] D. N. Reshef, Y. A. Reshef, H. K. Finucane, S. R. Grossman, G. McVean, P. J. Turnbaugh, E. S. Lander, M. Mitzenmacher, and P. C. Sabeti, "Detecting novel associations in large data sets," *Science*, vol. 334, no. 6062, pp. 1518–1524, 2011.

[68] 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.

[69] K. Tanaka, T. Toda, G. Neubig, S. Sakti, and S. Nakamura, "A hybrid approach to

electrolaryngeal speech enhancement based on noise reduction and statistical excitation generation," *IEICE Transactions on Information and Systems*, vol. E97-D, no. 6, pp. 1429–1437, Jun. 2014.

[70] S. Kang and H. Meng, "Statistical parametric speech synthesis using weighted multi-distribution deep belief network," in *Proc. INTERSPEECH*, Singapore, Sep. 2014, pp. 1959–1963.

[71] H. Zhang, T. Xu, H. Li, S. Zhang, X. Huang, X. Wang, and D. Metaxas, "StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks," *arXiv*, vol. abs/1612.03242, 2016.

[72] X. Yin, M. Lei, Y. Qian, F. K. Soong, L. He, Z.-H. Ling, and L.-R. Dai, "Modeling f0 trajectories in hierarchically structured deep neural networks," *Speech Communication*, vol. 76, pp. 82–92, 2016.

[73] T. Kaneko, H. Kameoka, N. Hojo, Y. Ijima, K. Hiramatsu, and K. Kashino, "Generative adversarial network-based postfilter for statistical parametric speech synthesis," in *Proc. ICASSP*, New Orleans, U.S.A., Mar. 2017, pp. 4910–4914.

[74] H. Zen, Y. Agiomyrgiannakis, N. Egberts, F. Henderson, and P. Szczepaniak, "Fast, compact, and high quality LSTM-RNN based statistical parametric speech synthesizer for mobile devices," in *Proc. INTERSPEECH*, California, U.S.A., Sep. 2016, pp. 2273–2277.

[75] K. Tokuda and H. Zen, "Directly modeling voiced and unvoiced components in speech waveforms by neural networks," in *Proc. ICASSP*, Shanghai, China, Mar. 2016, pp. 5640–5644.

[76] Y. Sagisaka, K. Takeda, M. Abe, S. Katagiri, T. Umeda, and H. Kawahara, "A large-scale Japanese speech database," in *ICSLP90*, Kobe, Japan, Nov. 1990, pp. 1089–1092.

[77] Y. Ohtani, T. Toda, H. Saruwatari, and K. Shikano, "Maximum likelihood voice conversion based on GMM with STRAIGHT mixed excitation," in *Proc. INTERSPEECH*, Pittsburgh, U.S.A., Sep. 2006, pp. 2266–2269.

[78] S. Takamichi, K. Kobayashi, K. Tanaka, T. Toda, and S. Nakamura, "The NAIST text-to-speech system for the Blizzard Challenge 2015," in *Proc. Blizzard Challenge workshop*, Berlin, Germany, Sep. 2015.

[79] Y. Saito, S. Takamichi, and H. Saruwatari, "Voice conversion using input-to-output highway networks," *IEICE Transactions on Information and Systems*, vol. E100-D, no. 8, pp. 1925–1928, 2017.

[80] T. Kaneko, S. Takaki, H. Kameoka, and J. Yamagishi, "Generative adversarial network-based postfilter for STFT spectrograms," in *Proc. INTERSPEECH*, Stockholm, Sweden, Aug. 2017, pp. 3389–3393.

[81] H. Saruwatari R. Sonobe, S. Takamichi, "JSUT corpus: free large-scale japanese speech corpus for end-to-end speech synthesis," *arXiv*, vol. abs/1711.00354, 2017.

[82] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, pp.

2121–2159, Jul. 2011.

[83] Y. Saito, S. Takamichi, and H. Saruwatari, "Training algorithm to deceive anti-spoofing verification for DNN-based speech synthesis," in *Proc. ICASSP*, New Orleans, U.S.A., Mar. 2017, pp. 4900–4904.

[84] 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," *arXiv*, vol. abs/1707.01670, 2017.

[85] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," in *Proc. ICML Deep Learning Workshop*, Lille, France, Jul. 2015.

[86] X. Wang, S. Takaki, and J. Yamagishi, "Investigating very deep highway networks for parametric speech synthesis," in *Proc. 9th ISCA Speech Synthesis Workshop*, California, U.S.A., Sep. 2016.

[87] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, Las Vegas, U.S.A., June 2016, pp. 770–778.

[88] T. Kitamura and M. Akagi, "Speaker individualities in speech spectral envelopes," *Journal of the Acoustical Society of Japan (E)*, vol. 16, no. 5, pp. 283–289, Sep. 1995.

[89] D. Erro, A. Moreno, and A. Bonafonte, "Voice conversion based on weighted frequency warping," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 5, pp. 922–931, Jul. 2010.

[90] J. Hout and A. Alwan, "A novel approach to soft-mask estmation and log-spectral enhancement for robust speech recognition," in *Proc. ICASSP*, Kyoto, Japan, Mar. 2012, pp. 4105–4108.

# Acknowledgements

I would like to express my deepest appreciation to Professor Hiroshi Saruwatari of the University of Tokyo, my thesis advisor, for his constant guidance and invaluable comments to this thesis. I have learned many valuable aspects of being a researcher from his attitude toward study.

I would also like to express my appreciation to Professor Kenji Yamanishi of the University of Tokyo and Lecturer Hideki Nakayama of the University of Tokyo, members of the thesis committee, for their valuable comments on the thesis.

I would especially like to express my gratitude to Research Assistant Professor Shinnosuke Takamichi of the University of Tokyo, for his continuous support and various advice. This work could not have been accomplished without his support. I would also like to thank Research Assistant Professor Daichi Kitamura, Associate Professor Shoichi Koyama, and Visiting Professor Kunio Kashino of the University of Tokyo, for their technical comments and lessons.

I want thank all members of System #1 Lab., Graduate School of Information Science and Technology, the University of Tokyo, for their encouragement. I also wish to express my deep gratitude to Ms. Motoko Kumai, Ms. Hiromi Ogawa, and Ms. Naoko Tanji, secretaries of our laboratory, for their kind help and support in all aspects of my research.

Finally, I would like to give my thanks to my family for all their support.

# A

# Voice Conversion Using Input-to-Output Highway Networks

## A.1 Introduction

In constructing acoustic models for VC, we can utilize not only techniques to alleviate the over-smoothing effect of converted speech parameters, but also input speech information since the input and output parameters are often in the same domain (e.g., cepstrum). This appendix proposes DNN-based VC using input-to-output highway networks. Although the typical DNN-based VC directly estimates a converted spectral parameter sequence, our architecture estimates it as the sum of input spectral parameters and weighted spectral differentials estimated through DNNs. The use of input speech parameters effectively alleviates the over-smoothing effect, and the weights of the spectral differentials effectively represent the characteristics of the spectral parameters.

## A.2 Proposed architecture

### A.2.1 Input-to-Output Highway networks for Voice Conversion

Highway networks [85, 86] are weighted skip-connections between layers, and they often connect hidden layers. Given that the input and output are often in the same domain (e.g., cepstrum) in VC, we propose a VC using highway networks connected from the input to output as follows:

$$\hat{\boldsymbol{y}} = \boldsymbol{x} + \boldsymbol{T}(\boldsymbol{x}) \circ \boldsymbol{G}(\boldsymbol{x}), \tag{A.1}$$

where $\circ$ is the Hadamard product. $\boldsymbol{T}(\cdot)$ is the transform gate of highway networks described as Feed-Forward neural networks. Each value of $\boldsymbol{T}(\boldsymbol{x})$ ranges from 0.0 to 1.0, and they represent time- and feature-varying weights of $\boldsymbol{G}(\boldsymbol{x})$. When $\boldsymbol{T}(\boldsymbol{x}) = \boldsymbol{0}$, input speech parameters are directly used as converted speech parameters, and when $\boldsymbol{T}(\boldsymbol{x}) = \boldsymbol{1}$, the architecture is equivalent to residual networks [87]. Therefore, the input speech parameters are strongly transformed by $\boldsymbol{G}(\cdot)$ when the value of the transform gate becomes

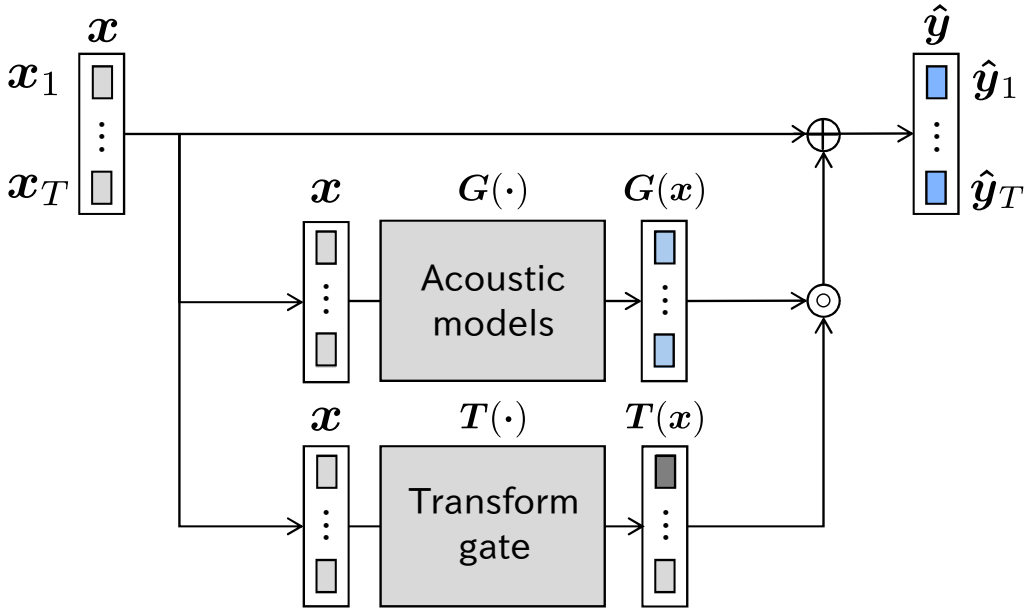


Fig. A.1: Voice conversion using input-to-output highway networks.

close to 1.0. Fig. A.1 shows the proposed architecture. The loss function for training is equal to the MGE loss shown in Eq. (2.4), and all model parameters of $\boldsymbol{T}(\cdot)$ and $\boldsymbol{G}(\cdot)$ are simultaneously estimated to minimize the loss function.

### A.2.2 Discussions

Since our architecture utilizes both input speech parameters and spectral differentials weighted by the transform gate, it efficiently alleviates over-smoothing of the converted speech parameters. Fig. A.2 shows scatter plots of the speech parameters. This figure plots pairs of mel-cepstral coefficients whose corresponding value of the transform gate is large (i.e., it is close to residual networks) or small (i.e., it is close to direct use of input speech parameters). We can see that our architecture alleviates distribution shrinkage better than Feed-forward neural networks in both cases.

The variation in spectral parameters between speakers strongly depends on not only the speaker pair but also the frequency band and phonetic environments. For instance, formant structures change more in the inter-gender case than in the intra-gender case, but the inter-speaker variation is small in the lower frequency bands within the same gender. On the other hand, inter-phoneme variation (intra-speaker variation) is large in the lower frequency bands [88]. Therefore, *golden* VC should avoid over-transformation (e.g., frequency warping [89]) when the input feature is close to the output feature and should apply a flexible transformation when the input is far from the output feature. The transform gates in Fig. A.1 can be interpreted as the variation and characteristics of the spectral parameters. Fig. A.3 shows examples of activation of the transform gates using mel-filter banks. This figure shows that $\boldsymbol{G}(\cdot)$ greatly transforms the spectral parameters in

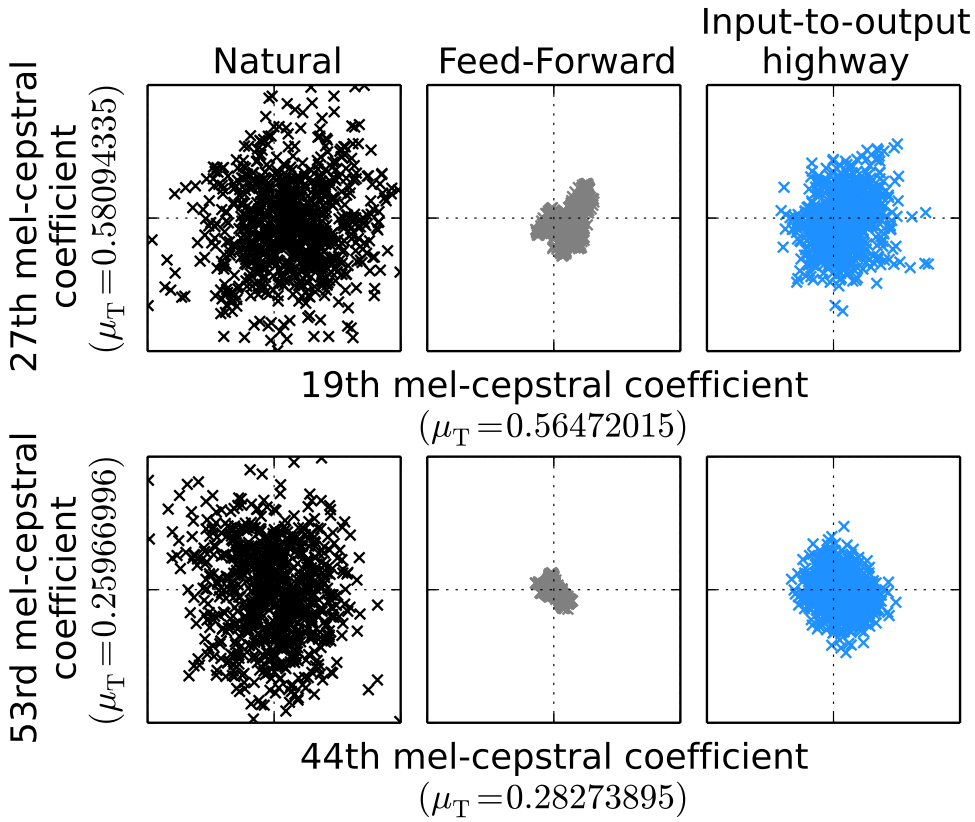


Fig. A.2: Scatter plots of speech parameters. $\mu_T$ denotes the value of the transform gate averaged over one utterance.

the high frequency band, since they strongly represent the characteristics of the speaker. Meanwhile, in male-to-male speaker conversion (Fig. A.3(a)), $\boldsymbol{G}(\cdot)$ does not transform the spectral parameters in the low frequency band as much as in the case of male-to-female speaker conversion (Fig. A.3(b)).

From another perspective, our architecture can be regarded as *soft* selection of features. The dimensionalities of the speech features (e.g., the numbers of mel-cepstral coefficients) are hyper-parameters for VC. For instance, the use of only the lower order of the mel-cepstrum makes the training robust while it degrades speech quality. On the other hand, the use of the rich orders improves speech quality but suffers from the randomness of the higher order of the mel-cepstrum. The former case corresponds to $\boldsymbol{T}(\boldsymbol{x}) = \boldsymbol{1}$ for the lower order and $\boldsymbol{T}(\boldsymbol{x}) = \boldsymbol{0}$ for the higher order. The latter case corresponds to $\boldsymbol{T}(\boldsymbol{x}) = \boldsymbol{1}$ for all orders. Whereas such a *hard* selection is often used, our architecture can utilize *soft* selection; i.e., each activation of $\boldsymbol{T}(\boldsymbol{x})$ varies from 0.0 to 1.0 depending on $\boldsymbol{x}$. Fig. A.4 shows examples of activation of the transform gates using mel-cepstral coefficients. We can see that the lower orders of the mel-cepstral coefficients, which are dominant in speaker conversion, tend to be strongly transformed by $\boldsymbol{G}(\cdot)$. On the other hand, the higher orders of the mel-cepstral coefficients tend to be not completetely ignored, but weakly converted.

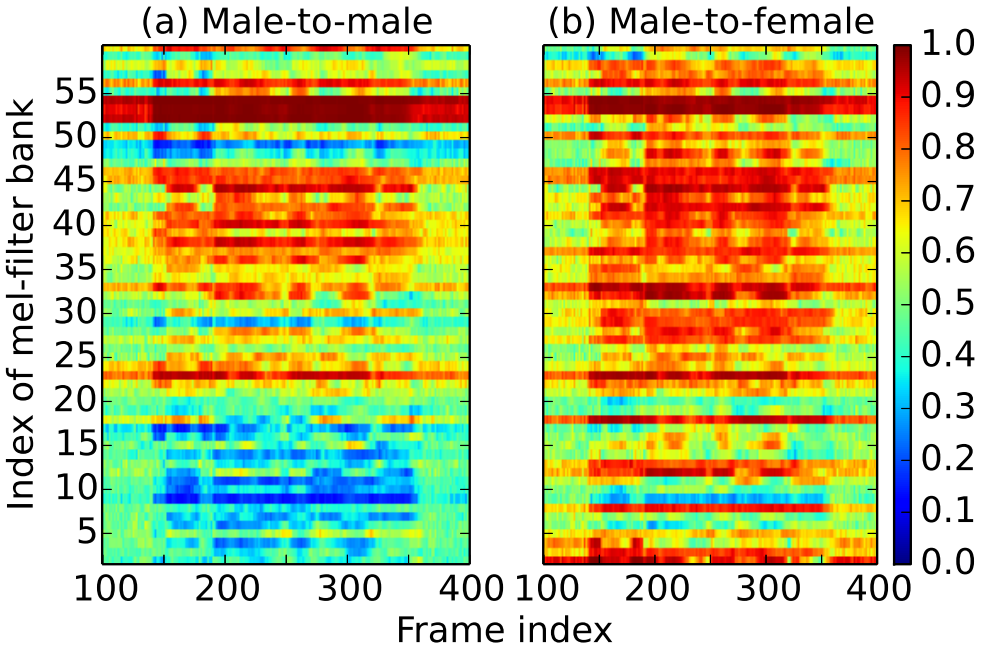


Fig. A.3: Examples of activation of transform gates using mel-filter banks.

Finally, the transform gate of our architecture is similar to adaptive soft-masking filtering [90] in speech enhancement. Hence, it is expected that knowledge can be shared between voice conversion and speech enhancement.

## A.3 Experimental Evaluation

### A.3.1 Experimental Conditions

We used speech data of two male speakers and one female speaker taken from the ATR Japanese speech database [76]. The speakers uttered 503 phonetically balanced sentences. We used 450 sentences (subsets A to I) for training and 53 sentences (subset J) for evaluation. Speech signals were sampled at a rate of 16 kHz, and the shift length was set to 5 ms. The 0th-through-59th mel-cepstral coefficients were used as the spectral parameter and $F_0$ and 5 band-aperiodicity [21, 77] were used as excitation parameters. The STRAIGHT analysis-synthesis system [24] was used for the parameter extraction and waveform synthesis. The 0th mel-cepstral coefficients of the input speech were directly used as those of the converted speech. To improve training accuracy, speech parameter trajectory smoothing [78] with a 50 Hz cutoff modulation frequency was applied to the spectral parameters in the training data. In the training phase, the spectral features were normalized to have zero-mean unit-variance, and the MGE training [39] was performed. We built DNNs for male-to-male and male-to-female conversion. The DNN architectures were Feed-Forward networks. The architecture included 3 $\times$ 512-unit Rectified Linear Unit (ReLU) [35] hid-

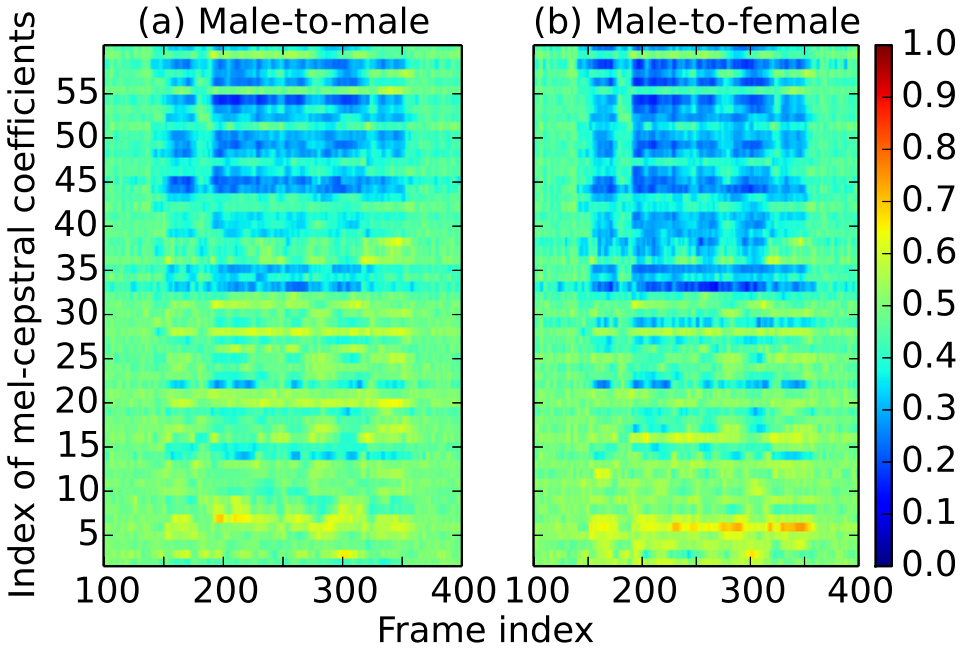


Fig. A.4: Examples of activation of transform gates using mel-cepstral coefficients.

den layers and a 118-unit linear output layer. The acoustic models output static and dynamic mel-cepstral coefficients (118-dim.) frame by frame. The transform gate only had a 59-unit input and 59-unit sigmoid output layers. We used AdaGrad [82] as the optimization algorithm, setting the learning rate to 0.01. $F_0$ was linearly transformed, and band-aperiodicity was not transformed.

To evaluate our architecture, we conducted a subjective evaluation of the converted speech quality and speaker individuality.

### A.3.2 Subjective Evaluation

In the subjective evaluation, we compared the proposed architecture (input-to-output highway) with the conventional one (Feed-Forward). A preference test (AB test) was conducted to evaluate the speech quality. We presented every pair of converted speech of the two architectures in random order, and we forced listeners to select speech samples that sounded like they had better quality. Similarly, an XAB test on the speaker individuality was conducted using natural speech as the reference, i.e., "X". Thirty listeners participated in each assessment of our crowd-sourced evaluation systems.

The results of the preference tests on speech quality and speaker individuality are shown in Fig. A.5 and Fig. A.6, respectively. We found that our architecture scored higher in both speech quality and speaker individuality than the conventional Feed-Forward neural network-based VC. Therefore, we demonstrated the effectiveness of our architecture.

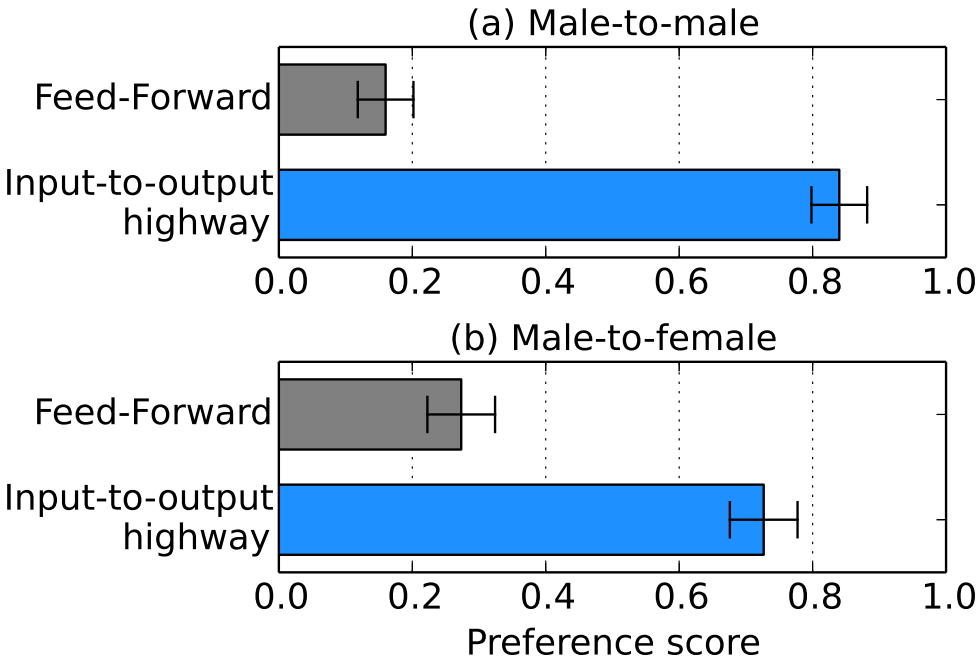


Fig. A.5: Preference scores of speech quality of converted speech with 95% confidence intervals (DNN-based VC using input-to-output highway networks).

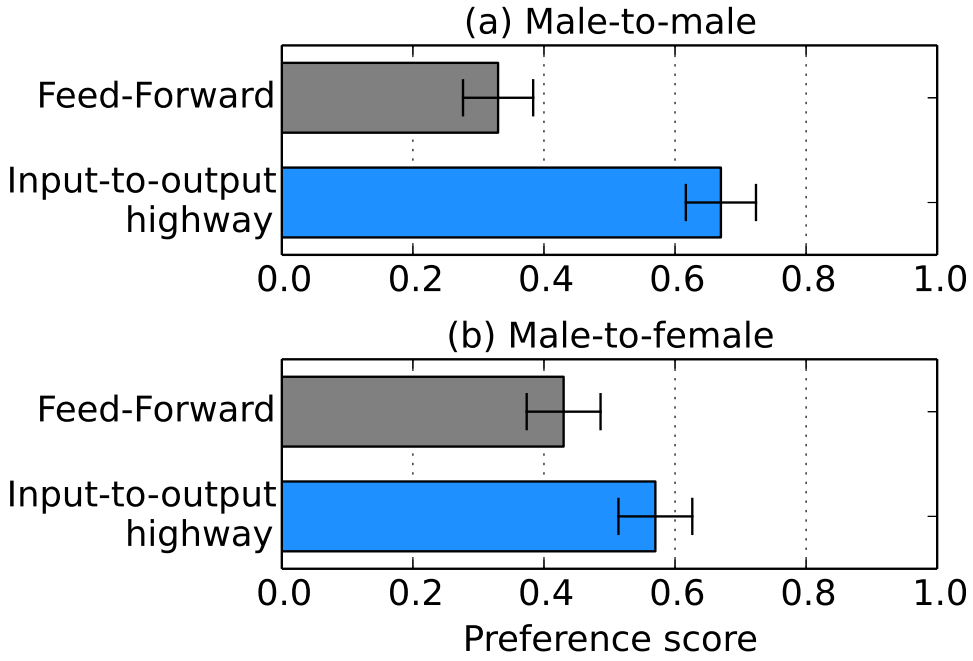


Fig. A.6: Preference scores of speaker individuality of converted speech with 95% confidence intervals (DNN-based VC using input-to-output highway networks).