博士論文

Statistical Speech Synthesis Based on Human's Speech Information Processing Abilities

（人間の音声情報処理能力に基づく統計的音声合成）

Yuki Saito

齋藤 佑樹

人間の音声情報処理能力に基づく統計的音声合成

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・変換・加工する技術である．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・パラ言語情報を変換する技術を音声変換という．音声合成は，人間とロボットの間での音声コミュニケーションを仲介するマンマシンインターフェースのみならず，音声バーチャルリアリティやエンターテインメント応用などを通じて，物理的制約を超えた音声表現を実現する技術である．本論文ではこれらの背景を踏まえ，多様な話者の音声を高品質に合成でき，かつ，合成された音声の話者性を直感的に制御可能な音声合成技術の確立を目指す．

本論文の主題である統計的音声合成は，統計的機械学習の枠組みに基づき，入力特徴量から合成音声パラメータを生成する音響モデルを学習する手法である．特に，音響モデルとして deep neural network (DNN) を用いる DNN 音声合成は，計算機性能の向上や大規模音声データベースの公開に恩恵を受け，入力特徴量から音声パラメータへの複雑な写像を学習可能な手法として広く研究されている．しかしながら，従来技術では，DNN 学習時の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，合成音声の品質が著しく劣化する．また，合成可能な話者は DNN 学習時に用いられたものに限定されるため，合成音声の話者性の多様性を高めることは困難である．さらに，合成音声の話者性を制御するために用いる従来の話者表現は，人間の主観的な話者知覚を無視しているため，多様な話者性の音声を合成可能な DNN 音声合成には不適切である．

本論文では，これらの問題点を解消するために，(1) 音声敵対過程を統合した高品質な音声合成法，(2) 音声認識過程を統合した多様な音声合成法，そして (3) 人間の音声知覚を導入した解釈性の高い話者表現学習法を提案する．これらの提案法は，音声合成を改善するために，人間の音声情報処理能力（敵対過程，認識過程，知覚過程）を活用するという着想に基づく．

(1) 音声敵対過程を統合した高品質な音声合成法では，generative adversarial network (GAN) を音声合成に導入し，高品質な音声を合成可能な DNN 音響モデルを学習する．この提案法は，人間の自然音声と合成音声を識別する能力（即ち，音声合成に対する敵対過程）を活用して DNN 音響モデルを学習するという着想に基づく．この提案法では，DNN 音響モデルと，自然音声と合成音声を識別する discriminator を交互に学習する．GAN の学習の目的関数は，生成データ分布と実データ分布の擬距離最小化とみなされるため，この提案法は，過

剰な平滑化を定量化するパラメトリックな統計的差異（音声パラメータ分布の 2 次モーメントなど）の補償により合成音声の品質を改善する従来法の理論拡張と解釈できる．また，この提案法における discriminator は，音声合成技術の悪用による音声なりすましを防ぐための anti-spoofing として解釈できる．したがって，anti-spoofing の知見を導入することで，より効率的な DNN 音響モデル学習が実現できる．本論文では，ボコーダ分析合成に基づくテキスト音声合成および音声変換のための GAN に基づく音声合成法を提案し，その有効性を評価する．その後，短時間フーリエ変換スペクトルを用いたより先進的なテキスト音声合成に向けてこの提案法を拡張する．実験的評価の結果から，この提案法によって合成音声の品質が有意に改善することを示す．

(2) 音声認識過程を統合した多様な音声合成法では，音声から発話内容と発話者を認識する DNN を導入し，高品質かつ多様な話者性を持つ音声を合成可能な DNN 音響モデルを学習する．この提案法は，人間の音声認識と話者認識（即ち，音声合成の逆過程）の能力を活用して DNN 音響モデルを学習するという着想に基づく．この提案法では，発話内容が明瞭で高品質な音声を合成するために，音声認識を用いて DNN 音響モデルを学習する．さらに，多様な話者性を持つ音声を合成するために，話者認識由来の連続的な話者表現を用いる．本論文では，変分オートエンコーダを用いた音声変換においてこの提案法の有効性を評価する．実験的評価の結果から，この提案法によって変換音声の有意な品質改善を達成しつつ，DNN 学習に用いていない未知話者の音声も合成可能になることを示す．

(3) 人間の音声知覚を導入した解釈性の高い話者表現学習法では，合成音声の話者性を直感的に制御可能な音声合成を実現するために，人間の主観的な音声知覚を導入して話者表現を学習する．この提案法は，人間の知覚を計算資源とみなし，解釈しやすい話者表現の学習に活用するという着想に基づく．この提案法では，まず，多数の評価者（音声の聞き手）による大規模主観評価により，多数話者間の知覚的な類似度のスコアを収集する．その後，このスコアを利用して，聞き手の話者知覚を高精度に再現する話者表現を学習する．このように学習された話者表現は，聞き手の話者知覚を強く反映するため，合成音声の話者性をより直感的に制御できるのみならず，未知話者の音声合成時の品質劣化に対する頑健性の向上が期待できる．本論文では，この提案法の有効性を (2) 音声認識過程を統合した多様な音声合成法において評価する．実験的評価の結果から，この提案法によって学習された話者表現の導入により，合成音声の品質と制御性が向上することを示す．

Abstract

Speech synthesis involves using a computer to synthesize, convert, and retouch human speech artificially. Text-to-speech (TTS) is a technique for synthesizing speech from text. Voice conversion (VC) is a technique for converting non-/para-linguistic information of input speech while retaining its linguistic information. These techniques provide a man-machine interface for mediating speech communication between humans and robots and enable voice expressions beyond human physical constraints through virtual reality and entertainment applications. This thesis aims to establish high-quality and intuitively controllable speech synthesis technology that can synthesize diverse speakers' voices.

Statistical speech synthesis, the subject of this thesis, trains an acoustic model that generates speech parameters from input features. Especially, deep neural network (DNN)-based speech synthesis has been widely studied because it can learn the complicated mapping from input features to output speech parameters benefiting from the improvement of computer performance and availability of large speech corpora. However, conventional DNN-based speech synthesis suffers from degrading synthetic speech quality caused by over-smoothing of generated speech parameters. It also has difficulty increasing speaker diversity in synthetic speech because the speakers' voices can be synthesized are limited to the speakers seen during the DNN training. Furthermore, conventional speaker representations for controlling speaker individuality in synthetic speech do not consider human subjective speaker perception. The use of such representations can worsen the controllability of DNN-based multi-speaker speech synthesis.

This thesis proposes three methods to overcome the above-described issues: 1) high-quality speech synthesis integrating a speech adversary process, 2) versatile speech synthesis integrating a speech recognition process, and 3) interpretable speaker representation learning introducing human speech perception. The core idea is to use human's speech information processing abilities for improving speech synthesis.

1) High-quality speech synthesis integrating a speech adversary process: This method introduces generative adversarial networks (GANs) to DNN-based speech synthesis for improving synthetic speech quality. The core idea is to use the human's adversarial ability that discriminates natural speech from synthetic speech. Two DNNs are alternatively trained with this method: an acoustic model and discriminator that distinguishes natural and synthetic speech. This GAN-like training leads to divergence (i.e., statistical difference) minimization between natural and generated speech parameters. Therefore,

this method can be interpreted as a theoretical extension of conventional methods for overcoming over-smoothing by reproducing parametric statistics of natural speech (e.g., second moments of speech parameter distribution). From a different perspective, the discriminator with this method can be regarded as anti-spoofing that prevents voice spoofing attacks. Accordingly, the anti-spoofing techniques can be introduced for efficient GAN-based acoustic model training. The effectiveness of this method is first evaluated for traditional TTS and VC using vocoder parameters. This method is then extended to advanced TTS using short-term Fourier transform spectra. The results indicate that this method significantly improves synthetic speech quality.

2) Versatile speech synthesis integrating a speech recognition process: This method involves integrating speech recognition and speaker classification into speech synthesis for increasing speaker diversity in synthetic speech while achieving higher quality. The core idea is to use the human's recognition ability that recognizes phonetic contents and speaker information of speech accurately. This method trains a DNN-based acoustic model using speech recognition. As a result, synthetic speech's phonetic content can be clarified, and high-quality speech can be synthesized. It also uses speaker-classification-derived speaker representations to diversify speaker individuality in synthetic speech. The effectiveness of this method is evaluated in VC using variational autoencoders. The results indicate that this method achieves significant improvement in converted speech quality and synthesizes unseen speakers' voices.

3) Interpretable speaker representation learning introducing human speech perception: This method involves introducing human subjective speech perception to learn speaker representations that enable intuitively controllable speech synthesis. The core idea is to regard the human's perception as computational resources for learning the interpretable speaker representations. The perceptual similarity scores among multiple speakers' voices are first collected with this method through a large-scale subjective evaluation involving many listeners. The scores are then used to learn speaker representations that consider the listeners' subjective speaker perception. Since the learned speaker representations strongly reflect listeners' perception, they are expected to enable highly controllable DNN-based multi-speaker speech synthesis. Such representations can also enhance the robustness against the quality degradation of unseen speakers' synthetic speech. The effectiveness of this method is evaluated with the second proposed method in this thesis. The results indicate that this method improves the quality and controllability of synthetic speech.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Speech is the most natural means for humans to communicate with each other. A speaker in speech communication transmits his/her intention, attitude, and emotion through the air as a speech waveform to a listener. Speech information processing aims to emulate such speech generation and recognition processes using a computer and assist in speech communication among humans and robots.

This thesis focuses on speech synthesis, a technology for artificially synthesizing, transforming, and retouching human speech by using a computer. This thesis covers two basic techniques in particular: text-to-speech (TTS) [1] and voice conversion (VC) [2]. Figure 1.1 illustrates the general frameworks of TTS and VC. TTS synthesizes speech from text and enables the development of a speech-based man-machine interface for computers, smartphones, and smart speakers. VC transforms non-/para-linguistic information of input speech while keeping its phonetic content unchanged and actualizes speech communication beyond human physical constraints. Figure 1.2 shows the typical applications of TTS and VC. These techniques can enrich human speech communication through language education [3, 4], virtual reality [5], and singing voice synthesis or conversion [6, 7, 8]. They can also remove barriers in human communication by developing speech translation [9, 10] and speaking aid [11, 12]. Therefore, the development of speech synthesis technology for synthesizing high-quality, versatile, and intuitively controllable speech will contribute to an advanced society where humans and computers cooperate.

One of the challenging aspects of speech synthesis research is to accurately synthesize human speech, which contains various information such as phonetic content and speaker identity. In general, even if the phonetic content is the same, human speech varies greatly

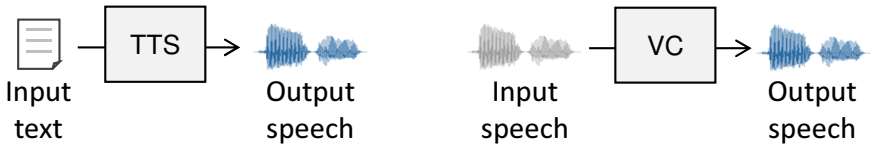

Fig. 1.1. Two speech synthesis techniques covered in this thesis, TTS and VC. Difference between TTS and VC is input information of speech synthesis systems.

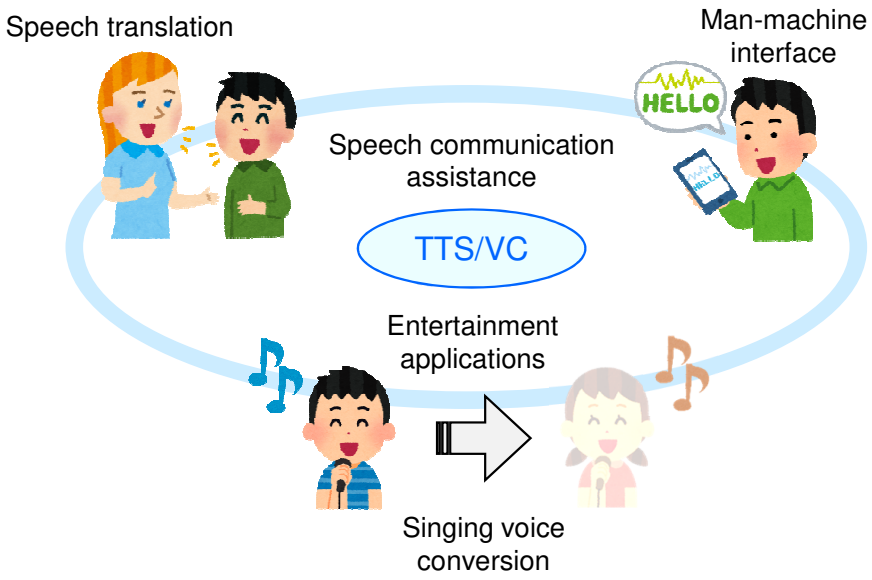

Fig. 1.2. Applications of speech synthesis technology

due to the speaker difference and intra-speaker variability (e.g., speaking style and emotion). Therefore, for example, TTS is required not only to read a given text accurately, but also to reproduce the identity and speaking style of the speaker with high fidelity. This challenge needs to be addressed for developing a speech synthesis technology that can synthesize high-fidelity speech and remove barriers to speech communication. However, a unified solution has not yet been established. This thesis focuses on monolingual speech synthesis and aims to establish a technology that can synthesize various speakers' voices with high quality, which can be the basis for technology to generate multilingual and diverse style voices. It also aims to develop a speech synthesis technology that can consider not only the speaker's voice reproduction but also the listener's voice perception to facilitate speech communication between humans and robots.

Speech synthesis methods can be classified into two categories: rule-based and corpus-based methods. Rule-based speech synthesis methods utilize some rules designed by experts, such as formant frequency, voicing, and noise levels, to synthesize speech artificially. Formant synthesis [13] is a well-known example of this method, which can

achieve a lightweight speech synthesis system with few parameters. However, the quality of speech synthesized by this method is far from that of natural speech due to the limited rules and the difficulty in synthesizing naturally-sounding consonants. On the other hand, corpus-based speech synthesis methods utilize a speech corpus containing human speech recordings to develop high-quality speech synthesis systems. Concatenative speech synthesis [14] and statistical one [15] are classified as this category. The former properly concatenate segments of speech waveforms or parameters stored in a speech corpus to synthesize speech. This method can achieve high naturalness of synthetic speech because natural speech's voice characteristics are preserved as much as possible. However, it requires a heavy footprint to store various speech recordings and has difficulty in controlling the voice characteristics of synthetic speech. Meanwhile, the latter train a statistical model that generates speech parameters from input features. This method's main advantage is the high controllability of synthetic speech with a relatively small footprint, thanks to its theoretically-grounded machine learning framework. However, synthetic speech quality tends to degrade due to many factors, such as improper speech parameterization and insufficient statistical modeling. Among these various methods, this thesis mainly focuses on statistical speech synthesis because it can approach the essence of human behavior that learns from and adapts to observed data.

A basic framework of statistical speech synthesis consists of two phases: training and synthesis [15]. In the training phase, input features, i.e., linguistic features in TTS and source speaker's speech parameters in VC, and output speech parameters are first extracted from a training speech corpus. An acoustic model is then trained to represent the relation between input features and output speech parameters. In the synthesis phase, input features are first extracted from the given input, i.e., arbitrary text in TTS and source speaker's arbitrary voice in VC. Speech parameters are then generated from the input features by using the trained model. Finally, a speech waveform is synthesized from the generated parameters. Through the training and synthesis phases, statistical speech synthesis offers a means to synthesize diverse speech and control voice characteristics in the synthetic speech better than concatenative TTS [14] and codebook-based VC [16]. Also, it can be extended to multi-speaker statistical speech synthesis [17, 18] that synthesizes multiple speakers' voices using a single acoustic model. Such multi-speaker acoustic modeling can widen the range of speech synthesis applications (e.g., speaker anonymization [19] and data augmentation [20]).

Deep neural network (DNN)-based speech synthesis [21, 22] involves DNNs [23] as acoustic models, one of the hottest topics in speech synthesis research. The main ad-

vantage of using DNNs is to synthesize high-fidelity speech better than traditional hidden Markov model (HMM)-based TTS [24] and Gaussian mixture model (GMM)-based VC [25]. The reason why is that DNNs can learn the complicated relation between input and output through a series of linear transformation and nonlinear activation functions. The improvement of computer performance and the availability of large speech corpora have further advanced DNN-based speech synthesis. Also, knowledge of other related DNN-based speech information processing, such as speech recognition [23, 26], speaker classification [27, 28], and anti-spoofing [29], can be introduced through the unified DNN training framework based on the backpropagation (BP) algorithm [30]. This thesis aims to develop high-quality, versatile, and intuitively controllable DNN-based speech synthesis, following these backgrounds.

1.2 Thesis Scope

1.2.1 Issues to Be Addressed

This section describes the issues with conventional DNN-based speech synthesis that are addressed in this thesis.

Low-quality Synthetic Speech Due to Over-smoothing

Although DNN-based speech synthesis can learn the complicated mapping from input features to speech parameters, the quality of synthetic speech degrades. A primary cause of quality degradation is over-smoothing [15, 31] of generated speech parameters. It removes the fine structures of natural speech parameters and makes the synthetic speech sound muffled. One can alleviate over-smoothing by training an acoustic model to consider the statistical difference between natural and generated speech parameters. For example, the probability distributions of natural speech parameters' statistics are first modeled in a parametric [25] or non-parametric [32] manner. Speech parameters are then generated or transformed using the distributions to reproduce the statistical properties of natural speech. A more effective approach is to use analytically derived features that quantify the degree of over-smoothing. A global variance (GV) [25] and modulation spectrum (MS) [33] of a speech parameter sequence are well-known examples of such features, and work as constraints in the acoustic model training [34, 35]. Nose and Ito [36] proposed a method of reducing the GV difference between natural and synthetic speech, assuming that their GVs follow the Gaussian distribution. Takamichi et al. [34] extended this idea to MS-based statistical property reproduction. However, quality degradation remains a

critical problem in both TTS and VC, regardless of the speech parameterization level, i.e., vocoder-derived parameters or short-term Fourier transform (STFT) spectra.

Limited Speaker Diversity in Synthetic Speech

A sufficiently large speech corpus that includes multiple speakers' speech utterances is required to train a DNN-based high-quality multi-speaker acoustic model. This requirement can be problematic in DNN-based parallel VC that trains DNNs using pairs of source and target speakers' speech utterances with the same phonetic content. However, it is unrealistic to record multiple speakers' parallel speech utterances. Non-parallel VC using variational autoencoders (VAEs) [37] has been proposed to solve this problem [38]. Although it can train a VAE-based acoustic model without preparing a parallel speech corpus, the converted speech quality degrades. One of the reasons for this quality degradation is over-regularization [39] of VAEs' latent variables expected to represent phonetic content of input speech. One can alleviate over-regularization by using a more informative prior distribution of the latent variables such as GMMs [40]. However, it is difficult to determine the number of clusters in a GMM prior because variations in the phonetic content are typically large. Also, the conventional VAE-based VC method cannot handle unseen speakers who are not included in training data, in spite of the VAEs' ability that can learn various speakers' voice characteristics.

Uninterpretable Speaker Representation

Speaker representations are additional input features for controlling speaker individuality of synthetic speech in DNN-based speech synthesis. Therefore, they play an essential role in developing intuitively controllable speech synthesis technology. Speaker representations for such technology should be designed considering how humans subjectively perceive the differences among speakers. The most straightforward speaker representation is a speaker code [41] that represents speaker individuality as a one-hot vector. However, human speaker perception is completely ignored because different speakers are orthogonal to each other in the speaker space constructed by speaker codes. Such speaker space prevents a user from finding his/her favorite voice characteristics intuitively. Also, a speaker code cannot define an unseen speaker's individuality due to its discrete nature, limiting the number of speakers' voices that can be synthesized.

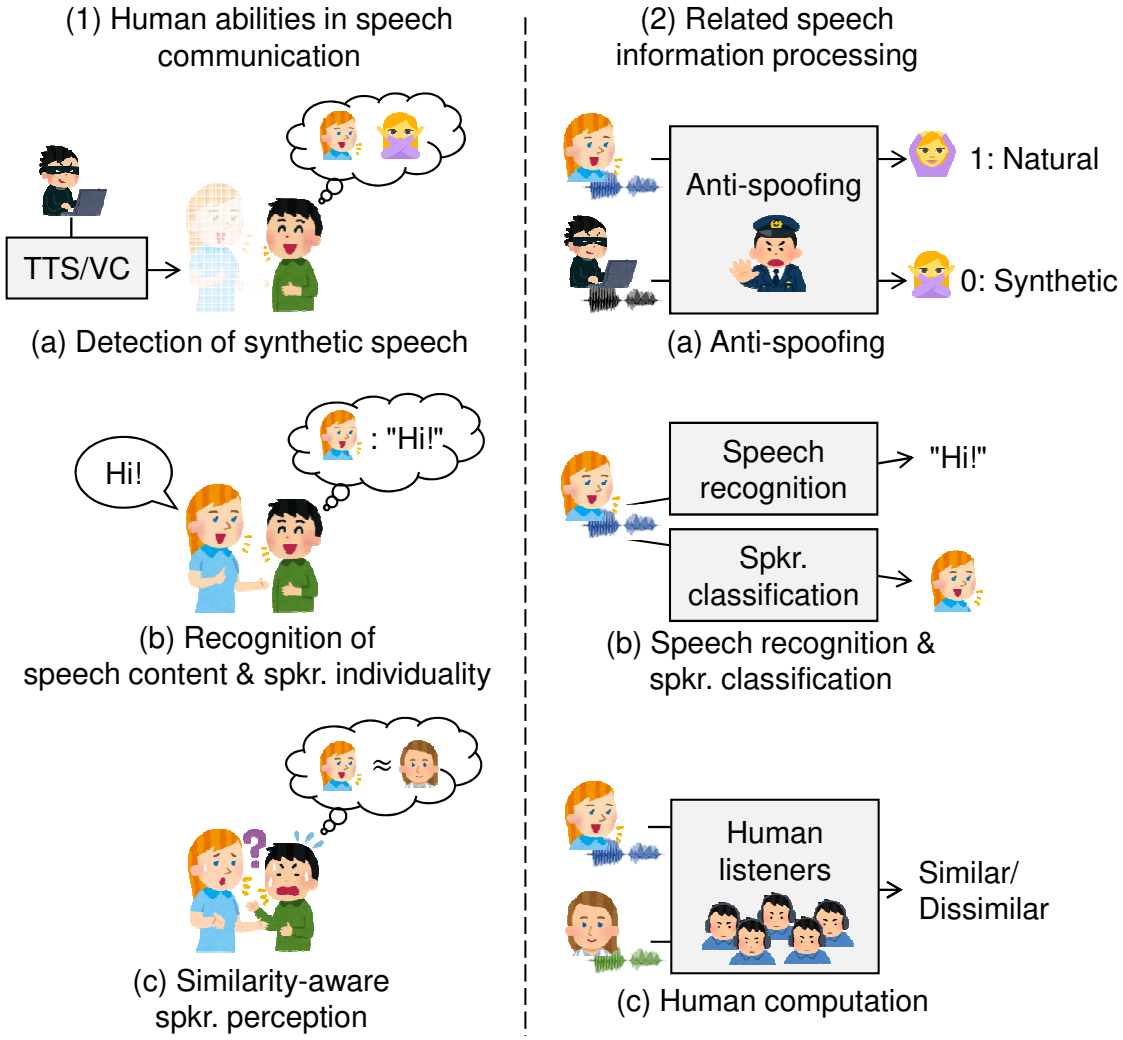

Fig. 1.3. Human abilities in speech communication (left) and related speech information processing (right)

1.2.2 Methods Proposed in This Thesis

This thesis proposes three methods to overcome these issues and further advance DNN-based speech synthesis technology: 1) high-quality speech synthesis integrating a speech adversary process, 2) versatile speech synthesis integrating a speech recognition process, and 3) interpretable speaker representation learning introducing human speech perception. The core idea of these methods is to utilize human's abilities related to speech information processing, i.e., adversary, recognition, and perception, to develop better statistical speech synthesis. Figure 1.3 contrasts the three abilities with related speech information processing. Humans can (1a) detect synthetic speech using their knowledge and experi-

ences, (1b) recognize the phonetic content or speaker individuality of speech accurately, and (1c) remember an unseen speaker's voice characteristics by associating them with memories. The first two can be substituted by data-driven machines created by humans as (2a) anti-spoofing and (2b) speech recognition or speaker classification, and the third is directly modeled on human responses through (2c) human computation.

High-quality Speech Synthesis Integrating Speech Adversary Process

A generative adversarial network (GAN)-based method is proposed to overcome the quality degradation caused by over-smoothing. The core idea is to use the human's adversarial ability that discriminates natural speech from synthetic speech. A discriminator that distinguishes natural speech parameters from generated ones is introduced with this method to emulate this ability. The DNN-based acoustic model is trained to fool the discriminator by generating speech parameters indistinguishable from natural ones. Since the objective of GANs is the minimization of divergence (i.e., the distribution difference) between natural and generated speech parameters, this method effectively alleviates over-smoothing. From a different perspective, the discriminator with this method can be interpreted as anti-spoofing, i.e., a technique to detect synthetic speech and prevent voice spoofing attacks. Accordingly, techniques and ideas concerning anti-spoofing can be introduced to improve synthetic speech quality further.

Versatile Speech Synthesis Integrating Speech Recognition Process

A VAE-based speech synthesis method is proposed to synthesize arbitrary speakers' voices with high quality. The core idea is to use the human's recognition ability that recognizes phonetic contents and speaker information of speech accurately. A DNN-based speech recognition model is introduced to train VAEs for speech synthesis. As a result, the synthetic speech's phonetic content is clarified, and high-quality synthetic speech is synthesized. Continuous speaker representations derived from a DNN-based speaker classification model are used to overcome the limitations of discrete speaker codes and increase speaker diversity.

Interpretable Speaker Representation Learning Introducing Human Speech Perception

A speaker representation learning method is proposed to increase the interpretability of speaker representations and enable controllable speech synthesis technology. The core idea is to regard the human's perception as computational resources for learning the interpretable speaker representations. This method incorporates human listeners into the

speaker representation learning framework based on human computation. A large-scale subjective scoring in terms of multiple speakers' perceptual similarity is first conducted involving a large number of listeners. DNNs are then trained for speaker representation learning considering the similarity scores. An active learning algorithm is proposed to reduce subjective scoring and DNN training costs.

1.3 Thesis Outline

The rest of this thesis is organized as follows (see also Fig. 1.4).

Chapter 2 briefly reviews the basic framework of DNN-based speech synthesis. The framework's four crucial factors are specifically described: 1) feature analysis, 2) acoustic modeling, 3) speech parameter generation, and 4) speech waveform synthesis.

Chapter 3 discusses the proposed GAN-based method to improve synthetic speech quality. The basic framework of GANs is first described. The introduction of GANs to train DNNs for speech synthesis is then discussed. From the divergence minimization perspective, the effects of the divergence in improving synthetic speech quality are investigated. Experimental evaluations are conducted to demonstrate this method's effectiveness in TTS and VC using vocoder-derived speech parameters. The results indicate that 1) this method can generate natural speech parameters regardless of its hyperparameter settings, and 2) Wasserstein GAN minimizing the earth-mover's distance works the best among other image-processing-related or speech-processing-related GANs in terms of improving synthetic speech quality.

Chapter 4 extends the proposed GAN-based method described in Chapter 3 to DNN-based speech synthesis using STFT spectra. A simple but effective GAN-based approach is proposed to overcome the difficulty in modeling high-dimensional and complicated amplitude spectra. Various frequency scales that are related to human speech perception are also introduced. The effectiveness of this method is evaluated in TTS using STFT spectra. The results indicate that 1) GANs using low-frequency-resolution amplitude spectra improve speech quality and work robustly against the settings of the frequency resolution and hyperparameters, 2) in comparison of low-, original-, and multi-frequency-resolution amplitude spectra, the use of low-frequency-resolution spectra works best to improve synthetic speech quality, and 3) the use of the inverse mel frequency scale for obtaining low-frequency-resolution amplitude spectra further improves synthetic speech quality.

Chapter 5 presents the proposed high-quality and versatile speech synthesis method

based on VAEs. The VAE-based speech parameter generation process is mathematically formulated to explicitly model the phonetic content and speaker individuality as latent variables. Non-parallel and many-to-many VC that can reproduce and transform arbitrary speaker's voice characteristics is established using VAEs. The trade-off between the number of seen speakers and dimensionality of continuous speaker representation with this method is also investigated. The effectiveness of this method is objectively and subjectively evaluated in VC. The results indicate that 1) the introduction of a DNN-based speech recognition model contributes to significant quality improvement in converted speech, 2) the use of continuous speaker representations achieves high-quality VC even if the source and target speakers' speech utterances are not used for the VAE training, and 3) high-dimensional speaker representation does not necessarily improve the converted speech quality, but a large number of seen speakers consistently does improve this.

Chapter 6 presents the proposed perceptual-similarity-aware speaker representation learning method for increasing the interpretability of speaker representations. The subjective scoring of perceptual speaker-pair similarity to obtain a perceptual speaker similarity matrix is first described. Methods of training DNNs for speaker representation learning using a loss function defined by the similarity matrix are then presented. An active learning algorithm to reduce the costs of scoring and training is finally introduced. This method's effectiveness is evaluated with the proposed VAE-based speech synthesis method described in Chapter 5. The results indicate that 1) the proposed speaker representation learning method learns speaker representations strongly correlated with perceptual similarity scores, 2) the representations improve synthetic speech quality better than conventional representations derived from a DNN-based speaker recognition model, and 3) the active learning algorithm achieves higher synthetic speech quality while reducing the costs of scoring and training.

Chapter 7 summarizes this thesis and mentions future directions.

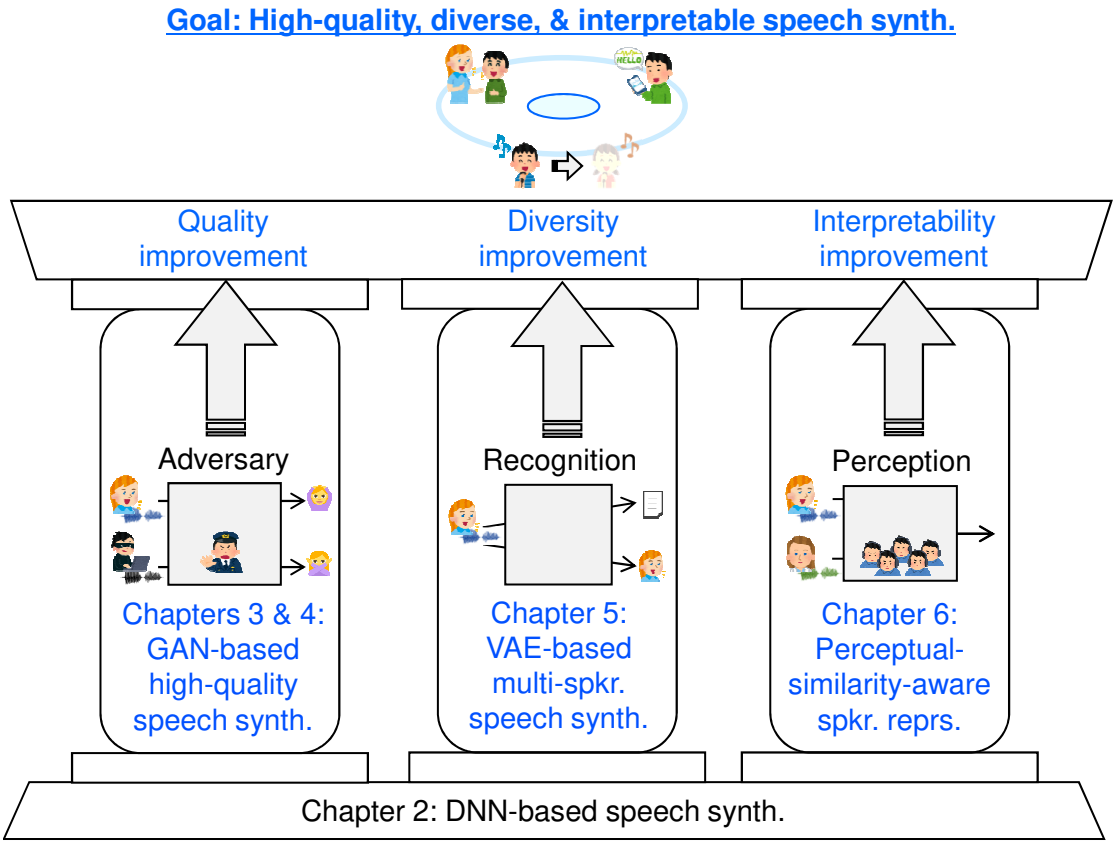

Fig. 1.4. Overview of this thesis. This thesis aims to develop high-quality, diverse, and interpretable DNN-based speech synthesis technology based on human's speech information processing abilities: adversary (Chapters 3 and 4), recognition (Chapter 5), and perception (Chapter 6).

Chapter 2

Statistical Speech Synthesis Using DNNs

2.1 Introduction

Statistical speech synthesis using DNNs, i.e., DNN-based speech synthesis, consists of two phases: training and synthesis. Figure 2.1 illustrates the flowcharts for DNN-based TTS and VC. In the training phase, input features (i.e., linguistic information of the given text in TTS or speech parameters of source speaker in VC) and output speech parameters are first extracted in the feature analysis step. A mapping from the input features to speech parameters is then modeled using DNNs in the acoustic modeling step. In the synthesis phase, input features are first extracted from the given input. Speech parameters are then generated by the trained DNNs using the extracted input features in the parameter generation step. A speech waveform is finally synthesized using the generated speech parameters in the speech waveform synthesis step. This chapter focuses on two primary techniques: vocoder-based and vocoder-free synthesis. DNN-based multi-speaker speech synthesis for synthesizing speech with versatile speaker individuality using a single acoustic model is also described.

This chapter is organized as follows (see also Fig. 2.2). Section 2.2 describes the feature analysis step, i.e., speech analysis for speech parameter extraction and text analysis for linguistic feature extraction. Section 2.3 explains the acoustic modeling step, including the DNN architecture design, specific TTS or VC modeling, the extension to multi-speaker modeling, and basic training objective functions. Section 2.4 gives an explanation of the speech parameter generation step. Section 2.5 explains the speech waveform synthesis step. Section 2.6 summarizes this chapter.

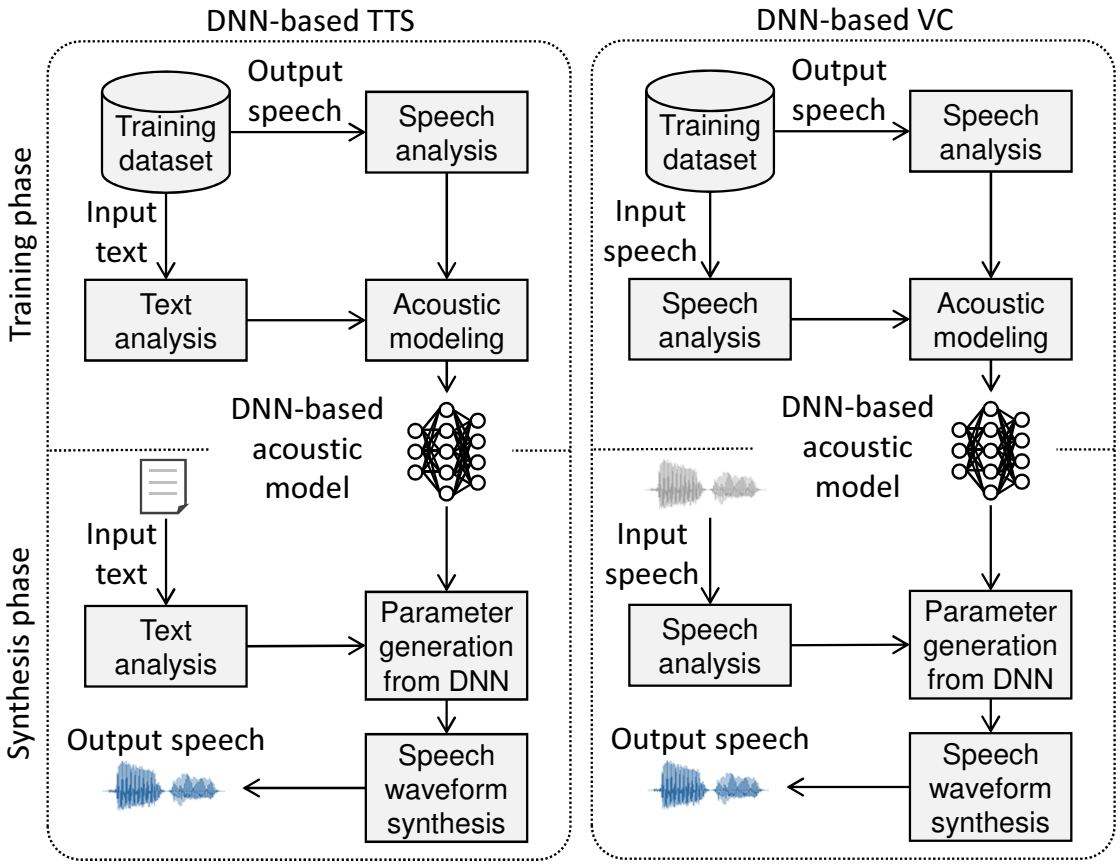

Fig. 2.1. Flowcharts for DNN-based TTS and VC. DNN-based acoustic model is trained to represent mapping from input features to output speech parameters.

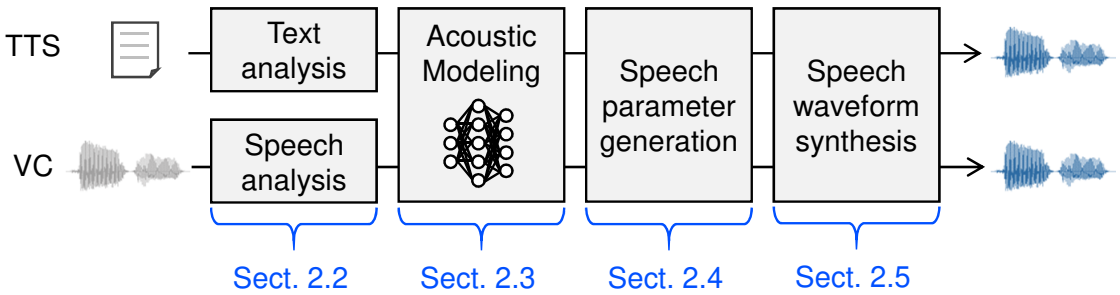

Fig. 2.2. Overview of Chapter 2

2.2 Feature Analysis

2.2.1 Speech Analysis

Figure 2.3 illustrates the conceptual diagram of speech analysis in statistical speech synthesis. STFT analysis is applied to a waveform for obtaining the time-frequency repre-

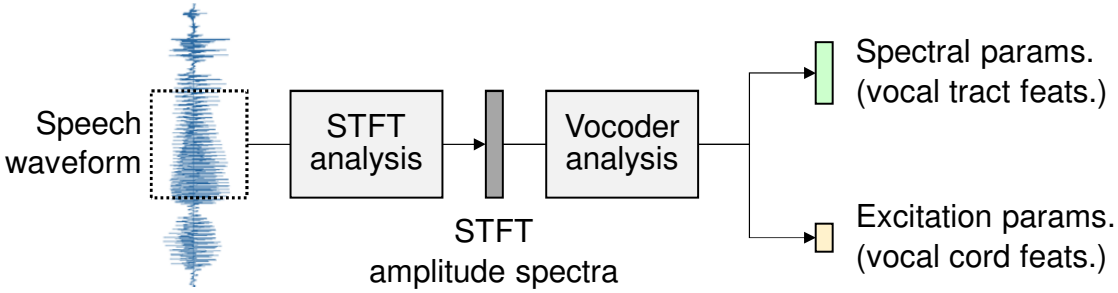

Fig. 2.3. Conceptual diagram of speech analysis. STFT analysis is first applied to speech waveform to obtain time-frequency representations of speech. Vocoder analysis is then carried out to decompose STFT amplitude spectra into spectral parameters and excitation parameters.

sentations of the waveform. Amplitude and phase spectra are given as the STFT analysis results. One can use these STFT spectra as the speech parameters to be modeled by a DNN-based acoustic model. However, it is difficult to predict the phase spectra using a statistical model due to the high randomness. Therefore, only the amplitude spectra are often considered as the speech parameters [42]. The amplitude spectra are further decomposed into spectral parameters (i.e., vocal tract features) and excitation parameters (i.e., vocal cord features). This decomposition is based on source-filter modeling [43] to obtain more controllable speech representations. The former and latter represent the spectral envelope (shown with a red dashed line in Fig. 2.4) and fine structure of the amplitude spectra, respectively. The excitation parameters are further decomposed into periodic factors typically represented as the fundamental frequency (F_0) and aperiodic factors [44].

As the dimensionality of spectral parameters tends to be high, a dimensionality reduction technique is applied to the parameters before the acoustic modeling. A common technique uses mel-cepstral coefficients [45] that consider the perceptual effects of human listening in lower frequency components for dimensionality reduction.

The difference between voiced regions (V) and unvoiced regions (U) must be considered in modeling F_0. Continuous F_0 modeling [46] was proposed to represent the F_0 parameters efficiently. It uses one-dimensional continuous values to represent the observed log F_0 and one-dimensional discrete values representing U/V (0 for U and 1 for V). The log F_0 values in unvoiced regions are estimated using SPLINE interpolation. Figure 2.5 shows an example of a continuous F_0 sequence and U/V labels.

Vocoder systems are typically utilized to extract the spectral and excitation parameters. A well-known example is the STRAIGHT [47] vocoder that can achieve high-quality

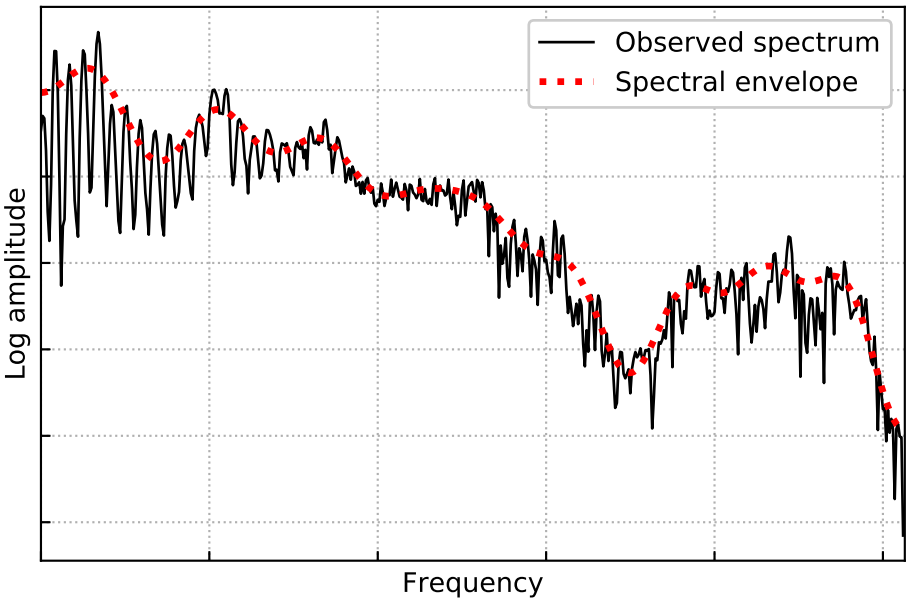

Fig. 2.4. Example of speech spectrum represented as product of spectral envelope and excitation parameters in frequency domain

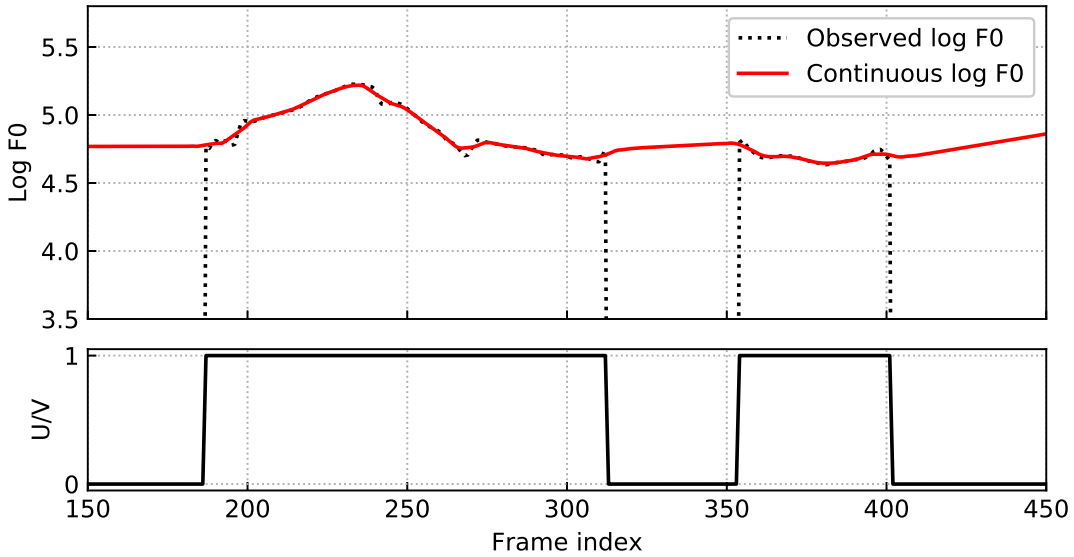

Fig. 2.5. Example of continuous F_0 modeling. Continuous F_0 sequence and U/V labels are separately modeled.

parameter extraction and speech synthesis. However, the deployment of speech synthesis systems using the STRAIGHT vocoder is limited due to its patent protection. Another example is the freely available WORLD [48, 49] vocoder.

2.2.2 Text Analysis

Text analysis is conducted to extract linguistic features from input text in TTS. This thesis focuses on Japanese TTS systems [50]. Japanese linguistic features consist of phoneme, accent type, word, part-of-speech, breath group, and sentence length, etc. A text analyzer, such as MeCab [51], is used to extract these features. The extracted linguistic features 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 are also included in the linguistic features.

2.2.3 Temporal Alignment

The lengths of the input features and output speech parameters are aligned for 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, considering the phoneme durations to align its length with the corresponding speech parameters. The Viterbi algorithm using HMMs is often used for phoneme duration prediction. Figure 2.6 shows an example of the feature alignment in TTS.

In VC, the source and target speech parameters' lengths are typically different; therefore, they need to be aligned before acoustic modeling. The dynamic time warping (DTW) algorithm [25] is used to align the lengths. Figure 2.7 shows a conceptual diagram of the DTW algorithm.

2.3 Acoustic Modeling

2.3.1 General Purpose

An acoustic model $\boldsymbol{G}(\cdot;\theta_{\mathrm{G}})$ parameterized by θ_{G} represents the mapping from an input feature sequence $\boldsymbol{x} = [\boldsymbol{x}_1^\top, \cdots, \boldsymbol{x}_t^\top, \cdots, \boldsymbol{x}_T^\top]^\top$ to output speech parameter sequence $\boldsymbol{y} = [\boldsymbol{y}_1^\top, \cdots, \boldsymbol{y}_t^\top, \cdots, \boldsymbol{y}_T^\top]^\top$, i.e., $\boldsymbol{y} = \boldsymbol{G}(\boldsymbol{x};\theta_{\mathrm{G}})$. The frame index and total frame length are denoted by t and T, respectively. A D_x-dimensional input feature vector and D_y-dimensional output speech parameter vector at frame t are given as $\boldsymbol{x}_t = [x_t(1), \cdots, x_t(D_x)]^\top$ and $\boldsymbol{y}_t = [y_t(1), \cdots, y_t(D_y)]^\top$, respectively. The goal with acoustic modeling is to estimate

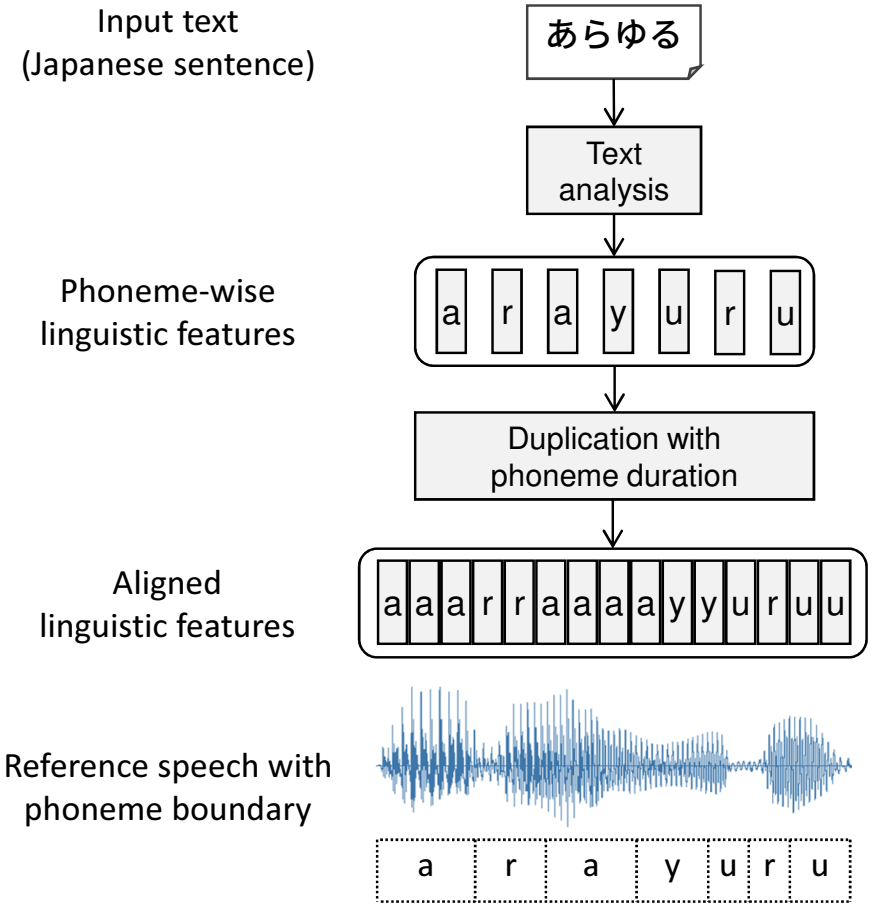

Fig. 2.6. Temporal alignment of features in TTS. Given phoneme boundary, each phoneme-wise linguistic feature is duplicated to align its length with corresponding speech parameters.

model parameters θ_{G} using a training dataset (i.e., a speech corpus) that includes pairs of input features and output speech parameters.

2.3.2 Static-dynamic Feature Modeling

Static-dynamic features of speech parameters are used for considering temporal continuity. 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. Dynamic features $\Delta\boldsymbol{y}_t$ and $\Delta\Delta\boldsymbol{y}_t$ are calculated as

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

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

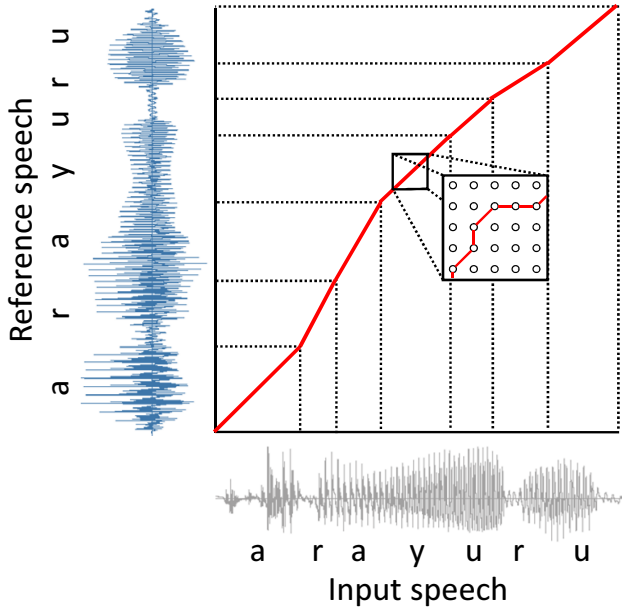

Fig. 2.7. DTW algorithm for feature alignment in VC. Phoneme boundaries of input and reference speech are superimposed for clear visualization.

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 [24]. Figure 2.8 shows the matrix computation to obtain the static-dynamic feature sequence.

2.3.3 DNN Architectures for Acoustic Modeling

A DNN is a class of artificial neural networks with more than one hidden layer between its input and output layers [23]. It provides a unified framework for acoustic modeling in both TTS and VC. This section focuses on three basic DNN architectures: Feed-Forward DNNs [21, 22], long-short term memory (LSTM) [52, 53], and VAEs [37].

Feed-Forward DNNs

Feed-Forward DNNs constitute the foundation of every DNN, which transform an input vector into an output vector through stacked nonlinear transformations. The layer-wise nonlinear transformations 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 the DNNs, respectively. A hidden vector at the lth layer $\boldsymbol{h}^{(l)}$ is calculated using the activation function as $\boldsymbol{h}^{(l)} = g^{(l)}(\boldsymbol{W}^{(l)}\boldsymbol{h}^{(l-1)} + \boldsymbol{b}^{(l)})$, where $\boldsymbol{W}^{(l)}$ and $\boldsymbol{b}^{(l)}$ denote a weight matrix of network connection and bias term at the lth layer, respectively. The input and output vectors of the DNNs correspond to $\boldsymbol{h}^{(0)}$ and $\boldsymbol{h}^{(L+1)}$, respectively. Figure 2.9 shows the

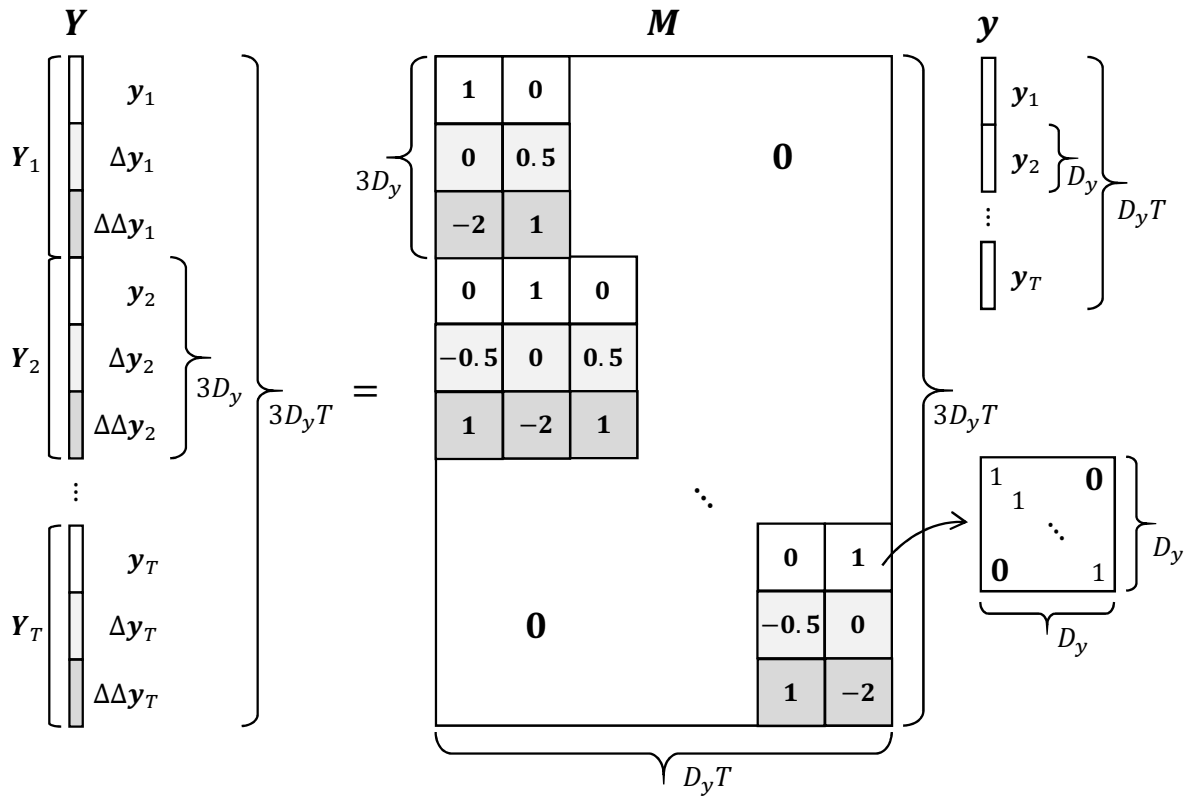

Fig. 2.8. Matrix computation to obtain static-dynamic feature sequence

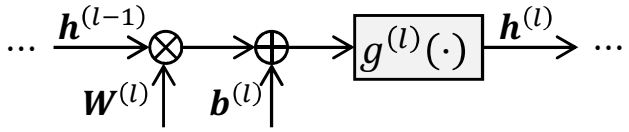

Fig. 2.9. Forward propagation procedure in Feed-Forward DNNs

forward propagation procedure to calculate $\boldsymbol{h}^{(l)}$ from $\boldsymbol{h}^{(l-1)}$ in the Feed-Forward DNNs.

The definition of the activation function plays a crucial role in the DNN framework. The following functions are often used for the hidden layers ($l = 1, \cdots, L$):

- Sigmoid: $\sigma(\boldsymbol{h}) = 1/(1 + e^{-\boldsymbol{h}})$
- Hyperbolic tangent (tanh): $\tanh(\boldsymbol{h}) = (1 - e^{-\boldsymbol{h}})/(1 + e^{-\boldsymbol{h}})$
- Rectified linear unit (ReLU) [54]: $\mathrm{ReLU}(\boldsymbol{h}) = \max\{0, \boldsymbol{h}\}$

The activation function for the output layer ($l = L + 1$) is designed in accordance with a task to be solved. The sigmoid function is used in binary classification tasks, e.g., speaker verification. The softmax function $\mathrm{Softmax}(\boldsymbol{h}) = e^{\boldsymbol{h}} / \sum e^{\boldsymbol{h}}$ is used in multi-class classification tasks, e.g., speech recognition and speaker classification. The linear function $\mathrm{Linear}(\boldsymbol{h}) = \boldsymbol{h}$ is used in regression tasks, e.g., TTS and VC.

The model parameters of Feed-Forward DNNs, i.e., the weight matrices and bias vectors of each hidden layer, are updated by supervised learning using the BP algorithm [30]. An output vector $\hat{\boldsymbol{y}}$ is first predicted from $\boldsymbol{x}$ through the DNNs. A defined loss function $L(\boldsymbol{y}, \hat{\boldsymbol{y}})$ is then computed with $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$. The model parameters are finally updated using the stochastic gradient descent (SGD) algorithm with the gradient $\boldsymbol{\nabla}_{\theta_{\mathrm{G}}} L(\boldsymbol{y}, \hat{\boldsymbol{y}})$.

LSTM

LSTM is one of the most popular recurrent neural network (RNN) architectures for sequence modeling. It has an LSTM block consisting of a memory cell $\boldsymbol{C}_t$, input gate $\boldsymbol{i}_t$, output gate $\boldsymbol{o}_t$, and forget gate $\boldsymbol{f}_t$ to learn long-short-term dependencies. Using a past hidden state $\boldsymbol{h}_{t-1}$, a past memory cell $\boldsymbol{C}_{t-1}$, and an input vector at current time step $\boldsymbol{x}_t$, the next $\boldsymbol{h}_t$ and $\boldsymbol{C}_t$ are calculated as follows:

$$\tilde{\boldsymbol{h}}_{t-1} = \mathrm{Concat}\left(\boldsymbol{h}_{t-1}, \boldsymbol{x}_t\right), \tag{2.3}$$

$$\boldsymbol{f}_t = \sigma\left(\boldsymbol{W}^{(f)} \tilde{\boldsymbol{h}}_{t-1} + \boldsymbol{b}^{(f)}\right), \tag{2.4}$$

$$\boldsymbol{i}_t = \sigma\left(\boldsymbol{W}^{(i)} \tilde{\boldsymbol{h}}_{t-1} + \boldsymbol{b}^{(i)}\right), \tag{2.5}$$

$$\boldsymbol{g}_t = \tanh\left(\boldsymbol{W}^{(g)} \tilde{\boldsymbol{h}}_{t-1} + \boldsymbol{b}^{(g)}\right), \tag{2.6}$$

$$\boldsymbol{C}_t = \boldsymbol{C}_{t-1} \circ \boldsymbol{f}_t + \boldsymbol{i}_t \circ \boldsymbol{g}_t, \tag{2.7}$$

$$\boldsymbol{o}_t = \sigma\left(\boldsymbol{W}^{(o)} \tilde{\boldsymbol{h}}_{t-1} + \boldsymbol{b}^{(o)}\right), \tag{2.8}$$

$$\boldsymbol{h}_t = \tanh\left(\boldsymbol{C}_t\right) \circ \boldsymbol{o}_t, \tag{2.9}$$

where $\mathrm{Concat}(\cdot)$ and $\circ$ denote the concatenating operation of given vectors and the Hadamard product (i.e., element-wise product), respectively. Information about the previous sequence is stored in $\boldsymbol{C}_{t-1}$, and the forget gate $\boldsymbol{f}_t$ controls the weight for $\boldsymbol{C}_{t-1}$. The input and output gates, $\boldsymbol{i}_t$ and $\boldsymbol{o}_t$, control the weights for $\boldsymbol{g}_t$ and $\tanh(\boldsymbol{C}_t)$, respectively. Figure 2.10 shows the forward propagation procedure to calculate $\boldsymbol{h}_t$ and $\boldsymbol{C}_t$ in LSTM.

The model parameters of LSTM, i.e., the weight matrices $\boldsymbol{W}^{(*)}$ and bias vectors $\boldsymbol{b}^{(*)}$, are updated by supervised learning using the backpropagation through time (BPTT) algorithm [55]. The inner loops of LSTM are first unfolded, then the algorithm is run to estimate the gradients used for the training.

VAEs

VAEs are probabilistic generative models that generate $\boldsymbol{x}$ from a latent variable $\boldsymbol{z}$. The model parameters of VAEs, θ, are estimated by maximizing the marginal likelihood of $\boldsymbol{x}$

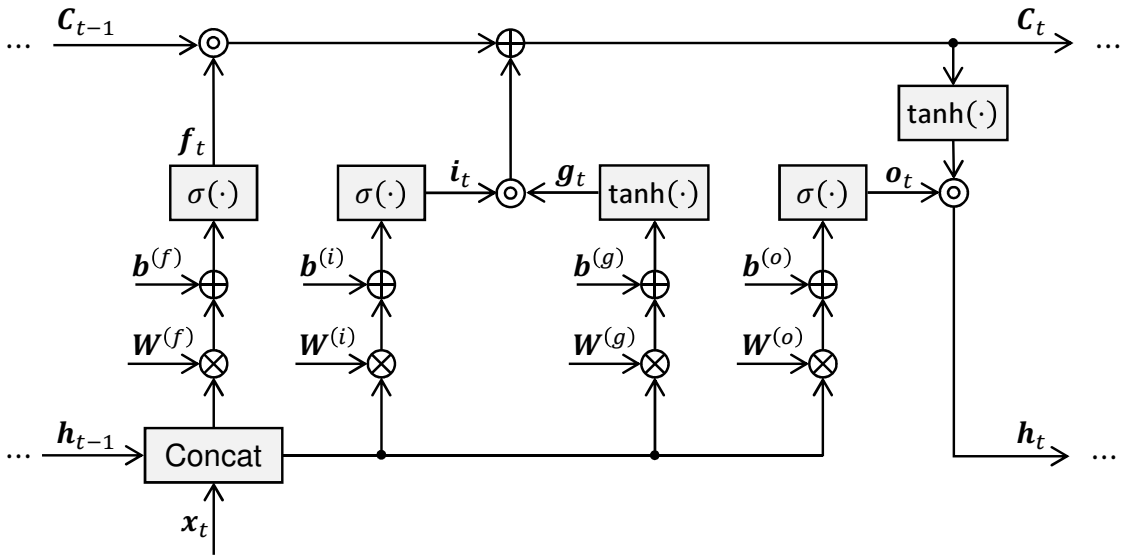

Fig. 2.10. Forward propagation procedure in LSTM. "Concat" denotes concatenating operation of given vectors, i.e., $\text{Concat}(\boldsymbol{h}_{t-1}, \boldsymbol{x}_t) = [\boldsymbol{h}_{t-1}^\top, \boldsymbol{x}_t^\top]^\top$.

defined as

$$p_\theta(\boldsymbol{x}) = \int p_\theta(\boldsymbol{x}|\boldsymbol{z})\, p_\theta(\boldsymbol{z})\, d\boldsymbol{z}, \tag{2.10}$$

where $p_\theta(\boldsymbol{z})$ is a prior distribution of $\boldsymbol{z}$. Since the integral in Eq. (2.10) is intractable, two DNNs are introduced: an encoder $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ and decoder $p_\theta(\boldsymbol{x}|\boldsymbol{z})$. The former and latter approximate the latent variable's true posterior $p_\phi(\boldsymbol{z}|\boldsymbol{x})$ and the data's true posterior $p_\theta(\boldsymbol{x}|\boldsymbol{z})$, respectively. Model parameters of the encoder and decoder are ϕ and θ, respectively. These model parameters are estimated to maximize the variational lower bound of the log likelihood defined as

$$\mathcal{L}(\theta, \phi; \boldsymbol{x}) = -\mathcal{D}_{\text{KL}}\left(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \,||\, p_\theta(\boldsymbol{z})\right) + \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})\right], \tag{2.11}$$

where $\mathcal{D}_{\text{KL}}(\cdot||\cdot)$ denotes the Kullback-Leibler (KL) divergence between two distributions. Both the encoder and decoder are assumed to represent the diagonal Gaussian distributions. These distributions' mean and covariance are estimated by the DNNs. The isotropic Gaussian distribution $\mathcal{N}(\boldsymbol{z}; \boldsymbol{0}, \boldsymbol{I})$ is typically adopted to $p_\theta(\boldsymbol{z})$ for obtaining the closed form of the KL term in Eq. (2.11). The reparameterization trick [37] is used for the BP algorithm to work. Figure 2.11 shows the forward propagation procedure to generate $\hat{\boldsymbol{x}}$ in the VAEs. The encoder first predicts the mean and variance of the latent variable's posterior probability, i.e., $\boldsymbol{\mu}_\phi$ and $\boldsymbol{\sigma}_\phi^2$, from $\boldsymbol{x}$. The reparameterization trick is then ap-

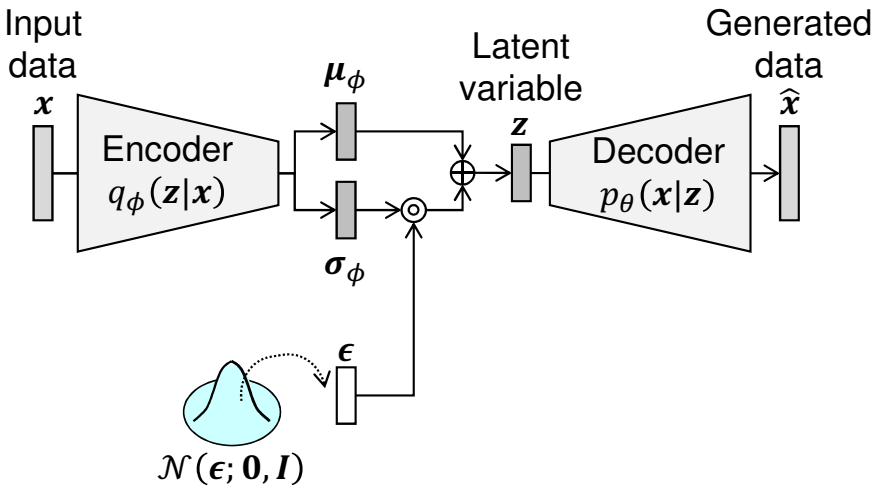

Fig. 2.11. Forward propagation procedure in VAEs. Based on reparameterization trick, VAE latent variable $\boldsymbol{z}$ is calculated as $\boldsymbol{z} = \boldsymbol{\mu}_\phi + \boldsymbol{\sigma}_\phi \circ \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon}$ is sampled from $\mathcal{N}(\epsilon; \boldsymbol{0}, \boldsymbol{I})$.

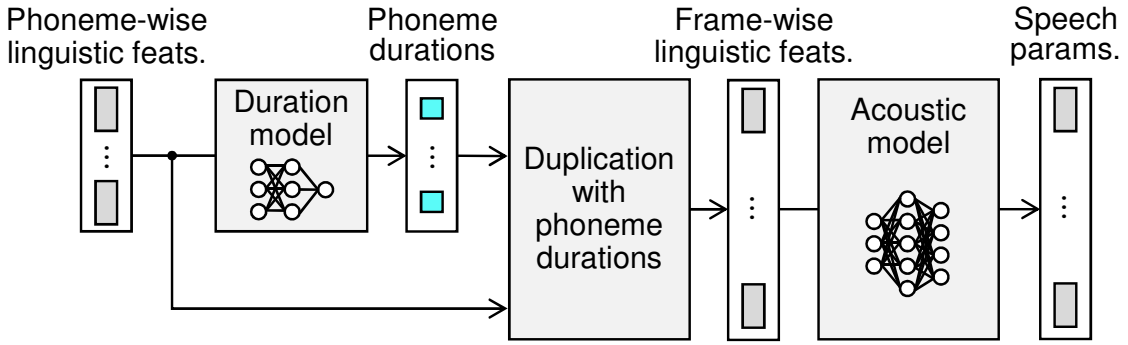

Fig. 2.12. Basic framework for DNN-based TTS

plied to emulate the latent variable sampling as $\boldsymbol{z} = \boldsymbol{\mu}_\phi + \boldsymbol{\sigma}_\phi \circ \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon}$ is sampled from $\mathcal{N}(\epsilon; \boldsymbol{0}, \boldsymbol{I})$. Finally, the decoder generates $\hat{\boldsymbol{x}}$ from $\boldsymbol{z}$.

2.3.4 DNN-based TTS

Basic Framework

Figure 2.12 shows the basic framework for DNN-based TTS. Two DNNs are prepared: one for the duration model and the other for the acoustic model. The former predicts phoneme durations from phoneme-wise linguistic features. The latter generates speech parameters from the linguistic features duplicated with the phoneme durations.

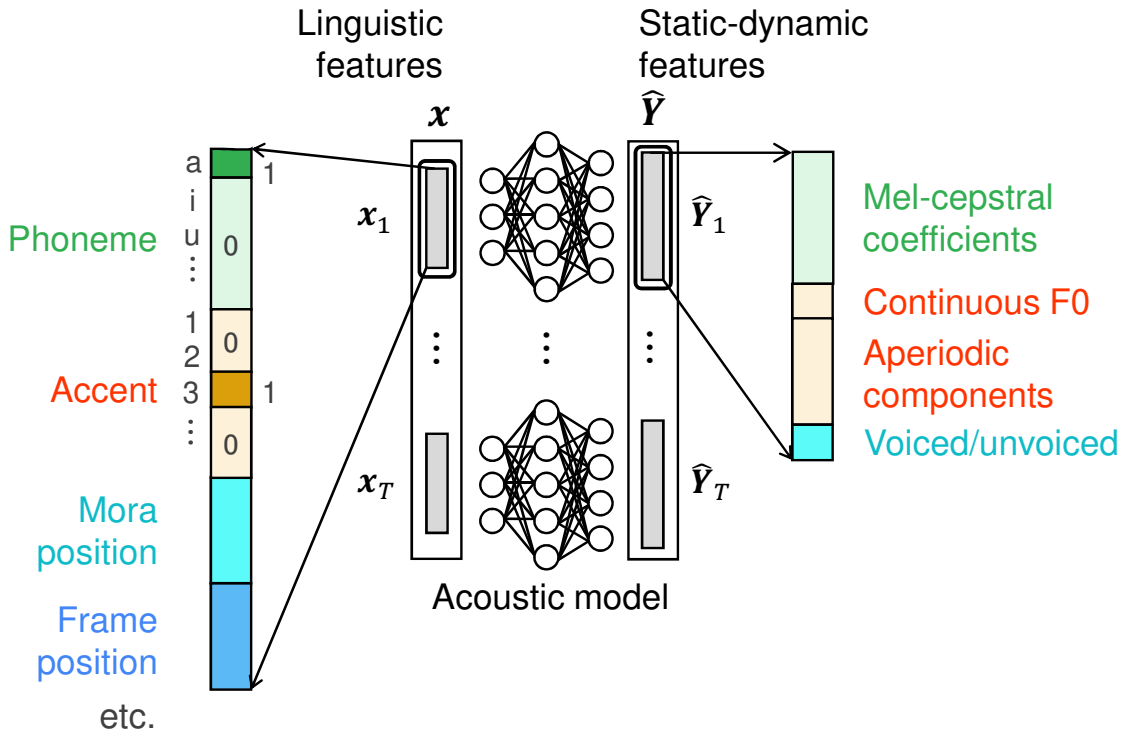

Fig. 2.13. DNN-based acoustic model in TTS using vocoder parameters. Frame-wise static-dynamic features are predicted from corresponding linguistic features.

DNN-based TTS Using Vocoder Parameters

An acoustic model in DNN-based TTS using vocoder parameters [21] generates a joint vector of U/V and static-dynamic features of mel-cepstral coefficients, continuous log F_0, and aperiodic components. Figure 2.13 shows the acoustic model representing the relation between linguistic features and speech parameters.

DNN-based Vocoder-free TTS Using STFT Spectra

DNN-based TTS using vocoder parameters works reasonably well. However, the use of vocoder-based parameterization sometimes causes buzziness in synthetic speech. One method for preventing this is DNN-based vocoder-free TTS using STFT spectra [42]. An acoustic model predicts a static feature sequence of STFT amplitude spectra from a joint vector of linguistic features, continuous F_0, and U/V. The use of F_0 parameters as input features is effective in predicting the harmonic information of amplitude spectra [42], and enables controlling the prosody of the synthetic speech intuitively. Accordingly, DNNs for F_0 parameter prediction need to be prepared in addition to the duration and acoustic models.

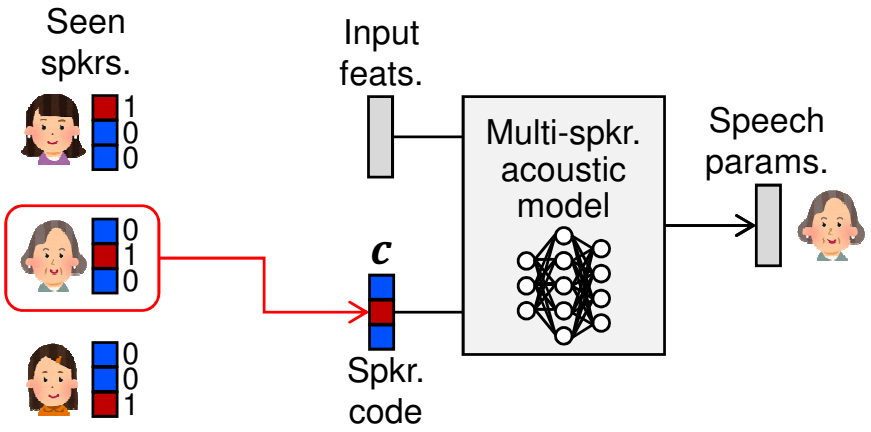

Fig. 2.14. DNN-based multi-speaker acoustic modeling using speaker code as speaker representation

2.3.5 DNN-based VC

An acoustic model in DNN-based VC predicts static-dynamic features of target speaker's mel-cepstral coefficients from those of source speaker's. F_0 is often linearly converted using the F_0 statistics of the source and target speech. The domains of the input and output features of acoustic modeling are the same in VC. Therefore, the acoustic model can be trained to represent the mapping from the input features to the difference between the two features. Kobayashi et al. [56, 57] proposed VC using spectral differentials that trains an acoustic model to represent $\boldsymbol{y} - \boldsymbol{x} = \boldsymbol{G}(\boldsymbol{x}; \theta_{\mathrm{G}})$, rather than $\boldsymbol{y} = \boldsymbol{G}(\boldsymbol{x}; \theta_{\mathrm{G}})$.

2.3.6 Multi-speaker Acoustic Modeling Using Speaker Codes

One can achieve multi-speaker acoustic modeling by conditioning DNNs for speech synthesis with speaker representations that control the speaker individuality of synthetic speech. Hojo et al. [41] proposed DNN-based multi-speaker acoustic modeling using a speaker code. The speaker code $\boldsymbol{c} = [c(1), \cdots, c(n), \cdots, c(N_{\mathrm{s}})]^\top$ represents the identity of one of the seen (i.e., pre-stored) N_{s} speakers. The ith speaker's identity $\boldsymbol{c}_i$ is defined as follows:

$$c_i(n) = \begin{cases} 1 & \text{if } n = i \\ 0 & \text{otherwise} \end{cases} \qquad (1 \leq n \leq N_{\mathrm{s}}). \tag{2.12}$$

Figure 2.14 shows a conceptual diagram of DNN-based multi-speaker acoustic modeling using speaker codes.

Similarly, Hsu et al. [38] proposed VAE-based multi-speaker acoustic modeling using speaker codes. They assumed that $\boldsymbol{z}$ is independent of the speaker individuality, and defined the marginal likelihood of $\boldsymbol{y}$ conditioned by $\boldsymbol{c}$ as follows:

$$p_\theta\left(\boldsymbol{y}|\boldsymbol{c}\right) = \int p_\theta\left(\boldsymbol{y}|\boldsymbol{z},\boldsymbol{c}\right) p_\theta\left(\boldsymbol{z}\right) d\boldsymbol{z}. \tag{2.13}$$

The speaker-dependent speech parameter generation process is modeled based on the conditional VAE [58] framework using a speaker-independent encoder $q_\phi(\boldsymbol{z}|\boldsymbol{y})$ and speaker-dependent decoder $p_\theta(\boldsymbol{y}|\boldsymbol{z},\boldsymbol{c})$. The variational lower bound of the log likelihood is defined as

$$\mathcal{L}\left(\theta,\phi;\boldsymbol{y},\boldsymbol{c}\right) = -\mathcal{D}_{\mathrm{KL}}\left(q_\phi\left(\boldsymbol{z}|\boldsymbol{y}\right) || \, p_\theta\left(\boldsymbol{z}\right)\right) + \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{y})}\left[\log p_\theta\left(\boldsymbol{y}|\boldsymbol{z},\boldsymbol{c}\right)\right]. \tag{2.14}$$

Since $\boldsymbol{z}$ is independent of the speaker individuality, it can be expected to represent the phonetic content of speech. Accordingly, VC can be conducted using the conditional VAEs by feeding the target speaker's code into the decoder. For instance, when converting source speech parameters into those of the jth speaker, $\boldsymbol{c}_j$ is fed into the decoder frame by frame. Note that parallel speech corpora are not required because the VAEs are trained in the same manner as in the auto-encoding process.

2.3.7 Loss Functions for DNN Training

The DNN-based acoustic model is trained to minimize a loss function calculated from the target speech parameters and output of the DNNs. The standard loss function is the mean squared error (MSE) between natural and generated speech parameters. The acoustic model in TTS or VC using vocoder parameters is trained to minimize the MSE between a natural static-dynamic feature sequence $\boldsymbol{Y}$ and that predicted by DNNs $\hat{\boldsymbol{Y}}$:

$$L_{\mathrm{MSE}}\left(\boldsymbol{Y},\hat{\boldsymbol{Y}}\right) = \frac{1}{T}\left(\hat{\boldsymbol{Y}}-\boldsymbol{Y}\right)^\top\left(\hat{\boldsymbol{Y}}-\boldsymbol{Y}\right). \tag{2.15}$$

The MSE is also used for the duration model training in TTS. Because the dynamic features of the STFT amplitude spectra are not considered to be modeled by the acoustic model in DNN-based vocoder-free TTS, $\boldsymbol{Y}$ and $\hat{\boldsymbol{Y}}$ in Eq. (2.15) are replaced with $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$, respectively,

Minimum generation error (MGE) training was proposed [59, 60] to consider the static-dynamic constraint of the speech parameters. The loss function in MGE training is

defined as the MSE between the natural and generated speech parameters after applying the maximum likelihood parameter generation (MLPG) algorithm [61]

$$\begin{aligned} L_{\mathrm{MGE}}\left(\boldsymbol{y}, \hat{\boldsymbol{y}}\right) &= \frac{1}{T}\left(\hat{\boldsymbol{y}}-\boldsymbol{y}\right)^{\top}\left(\hat{\boldsymbol{y}}-\boldsymbol{y}\right) \\ &= \frac{1}{T}\left(\boldsymbol{R}\hat{\boldsymbol{Y}}-\boldsymbol{y}\right)^{\top}\left(\boldsymbol{R}\hat{\boldsymbol{Y}}-\boldsymbol{y}\right), \end{aligned} \tag{2.16}$$

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

$$\boldsymbol{R} = \left(\boldsymbol{M}^{\top}\boldsymbol{\Sigma}^{-1}\boldsymbol{M}\right)^{-1}\boldsymbol{M}^{\top}\boldsymbol{\Sigma}^{-1}. \tag{2.17}$$

A $3D_yT$-by-$3D_yT$ covariance matrix $\boldsymbol{\Sigma} = \mathrm{diag}[\boldsymbol{\Sigma}_1, \cdots, \boldsymbol{\Sigma}_t, \cdots, \boldsymbol{\Sigma}_T]$ consists of $\boldsymbol{\Sigma}_t$, i.e., a $3D_y$-by-$3D_y$ covariance matrix at frame t. The covariance matrix $\boldsymbol{\Sigma}$ is separately estimated using a training dataset. The gradient $\nabla_{\hat{Y}} L_{\mathrm{MGE}}(\boldsymbol{y}, \hat{\boldsymbol{y}})$ is given as $\boldsymbol{R}^{\top}(\hat{\boldsymbol{y}}-\boldsymbol{y})/T$ [60].

The loss function in VAE-based acoustic modeling [38] is defined as the negative log likelihood of the VAEs, i.e., $\mathcal{D}_{\mathrm{KL}}\left(q_\phi\left(\boldsymbol{z}|\boldsymbol{y}\right) || \, p_\theta\left(\boldsymbol{z}\right)\right) - \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{y})}\left[\log p_\theta\left(\boldsymbol{y}|\boldsymbol{z}, \boldsymbol{c}\right)\right]$. The first term $\mathcal{D}_{\mathrm{KL}}\left(q_\phi\left(\boldsymbol{z}|\boldsymbol{y}\right) || \, p_\theta\left(\boldsymbol{z}\right)\right)$ can be regarded as the regularization term on $\boldsymbol{z}$. The KL divergence can be evaluated in closed form with the assumption of the Gaussian prior:

$$\begin{aligned} \mathcal{D}_{\mathrm{KL}}\left(q_\phi\left(\boldsymbol{z}|\boldsymbol{y}\right) || \, p_\theta\left(\boldsymbol{z}\right)\right) &= \mathcal{D}_{\mathrm{KL}}\left(\mathcal{N}\left(\boldsymbol{z}; \boldsymbol{\mu}_\phi, \mathrm{diag}\left[\boldsymbol{\sigma}_\phi^2\right]\right) || \mathcal{N}\left(\boldsymbol{z}; \boldsymbol{0}, \boldsymbol{I}\right)\right) \\ &= \frac{1}{2}\sum_{d=1}^{D_z}\left(1 + \log\sigma_\phi^2\left(d\right) - \mu_\phi^2\left(d\right) - \sigma_\phi^2\left(d\right)\right), \end{aligned} \tag{2.18}$$

where D_z denotes the dimensionality of $\boldsymbol{z}$. The second term $-\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{y})}\left[\log p_\theta\left(\boldsymbol{y}|\boldsymbol{z}, \boldsymbol{c}\right)\right]$ can be calculated as the reconstruction error of the VAEs, i.e., $L_{\mathrm{MSE}}(\boldsymbol{y}, \hat{\boldsymbol{y}})$ [37].

2.4 Speech Parameter Generation

2.4.1 MLPG algorithm

Speech parameters are generated from the trained acoustic model. In DNN-based TTS using vocoders, the phoneme durations of the given linguistic features are first predicted using the trained duration model. The static-dynamic feature sequence of the speech parameters is then predicted using this model. The MLPG algorithm is finally applied to obtain the static features of the speech parameters. Similar steps are taken in DNN-based vocoder-free TTS, except for applying the MLPG algorithm because the acoustic model predicts static features of the STFT amplitude spectra directly. In DNN-based VC,

the static-dynamic feature sequence of the target speech parameters is predicted using the source speech parameters, and the MLPG algorithm is applied to obtain the static features of the target speech parameters.

2.4.2 GV Compensation

Although the generated speech parameters after the MLPG algorithm are temporally smoothed, their fine structures tend to vanish due to over-smoothing, which considerably degrades synthetic speech quality. One way to prevent the fine structures from vanishing is to reproduce the statistics of the natural speech. GV compensation [62, 63] is a commonly used technique for improving synthetic speech quality. A GV is defined as the second moment of the speech parameter sequence. A D_y-dimensional GV vector of $\boldsymbol{y}$ is calculated using

$$\boldsymbol{v}\left(\boldsymbol{y}\right) = \left[v\left(1\right), \cdots, v\left(d\right), \cdots, v\left(D_y\right)\right]^\top, \tag{2.19}$$

$$v\left(d\right) = \frac{1}{T}\sum_{t=1}^{T}\left(y_t\left(d\right) - \left\langle y\left(d\right)\right\rangle\right)^2, \tag{2.20}$$

$$\left\langle y\left(d\right)\right\rangle = \frac{1}{T}\sum_{t=1}^{T} y_t\left(d\right). \tag{2.21}$$

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

$$\hat{y}_t^{(\mathrm{GV})}\left(d\right) = \sqrt{\frac{\mu^{(\mathrm{GV})}\left(d\right)}{\hat{\mu}^{(\mathrm{GV})}\left(d\right)}}\left\{\hat{y}_t\left(d\right) - \left\langle\hat{y}\left(d\right)\right\rangle\right\} + \left\langle\hat{y}\left(d\right)\right\rangle, \tag{2.22}$$

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

2.5 Speech Waveform Synthesis

This section describes two types of speech waveform synthesis: vocoder-based synthesis and vocoder-free synthesis.

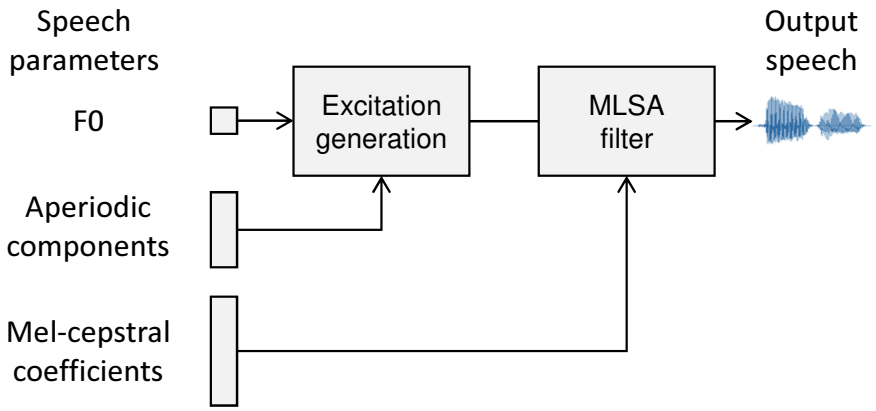

Fig. 2.15. Speech synthesis using MLSA filter. Excitation signal is first generated from F_0 and aperiodic components, then MLSA filter is applied to it for synthesizing speech waveform.

2.5.1 Vocoder-based Synthesis

Mel-log Spectrum Approximation (MLSA) Filter

The synthetic speech waveform in DNN-based speech synthesis using vocoder parameters is synthesized from the generated speech parameters using a synthesis filter such as the MLSA filter [64]. Figure 2.15 illustrates the speech synthesis process using the MLSA filter.

2.5.2 Vocoder-free Synthesis

The analysis of the excitation parameters often incurs errors such as the U/V decision error. Therefore, several vocoder-free synthesis methods have been developed for avoiding errors and improving speech quality.

Spectral Differentials Filter

The converted speech waveform in VC using spectral differentials [56, 57] is synthesized by applying a spectral differential filter to the input speech waveform. Figure 2.16 illustrates the VC process using the spectral differentials.

Phase Reconstruction from Spectral Amplitudes

Only the amplitude spectra are modeled by the acoustic model in vocoder-free TTS using STFT spectra. Therefore, phase spectra are reconstructed using the Griffin and Lim algorithm [65] with the predicted amplitude spectra. Let $\mathrm{y}(n)$ be a speech waveform

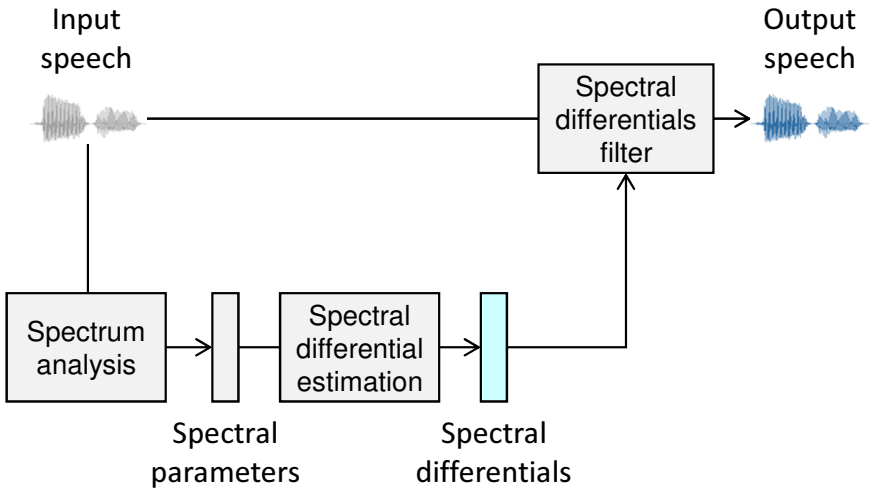

Fig. 2.16. VC using spectral differentials. Input speech waveform is converted using estimated spectral differentials filter.

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 amplitude spectra $y_t(d)$:

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

7: **end for**

sample at time index n. The phase information for the given STFT amplitude spectra $y_t(d)$ is reconstructed in accordance with Algorithm 2.1. This algorithm can synthesize a speech waveform without the vocoding process. Figure 2.17 illustrates the TTS process using STFT spectra.

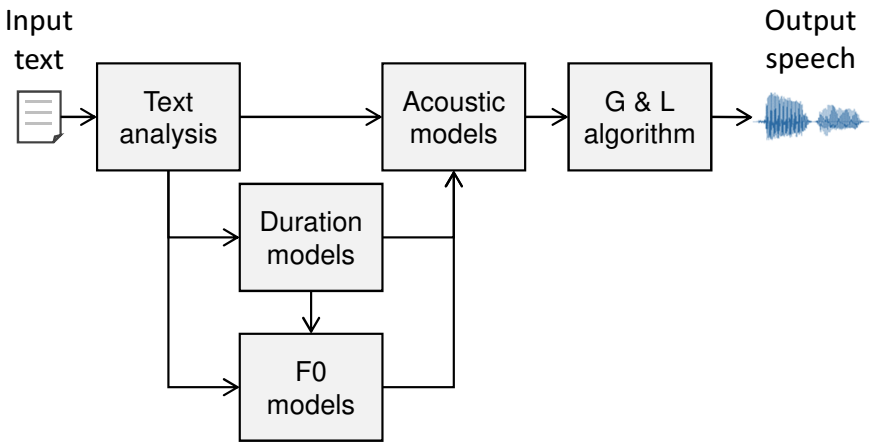

Fig. 2.17. Vocoder-free TTS process using STFT spectra. "G & L" indicates "Griffin and Lim."

2.6 Summary

This chapter reviewed the basic framework of statistical speech synthesis using DNNs as the acoustic models. The framework consists of two phases: training and synthesis. The framework's four crucial factors: 1) feature analysis, 2) acoustic modeling, 3) speech parameter generation, and 4) speech waveform synthesis, were described. The DNN-based acoustic model plays an essential role in representing the complicated relation between input features and speech parameters. Feed-Forward DNNs, LSTM, and VAEs are often used for acoustic modeling. The model parameters of DNNs are estimated using the BP (or BPTT) algorithm to minimize the defined loss function.

Chapter 3

Vocoder-based Statistical Speech Synthesis Using GANs

3.1 Introduction

This chapter proposes a novel algorithm using GANs to train a DNN-based acoustic model and prevent over-smoothing of generated speech parameters. Figure 3.1 shows a conceptual diagram of the proposed algorithm. GANs consist of two DNNs: a discriminator to distinguish natural and generated samples, and a generator to generate samples that fool the discriminator. A new training criterion for the acoustic model is defined based on the GAN framework. The criterion is the weighted sum of the conventional MGE and GAN-derived adversarial loss. The adversarial loss makes the discriminator recognize the generated speech parameters as natural. Since the objective of GANs leads to minimize divergence (i.e., distribution difference) between natural and generated speech parameters, this algorithm can alleviate over-smoothing and improve synthetic speech quality. Moreover, this algorithm can be regarded as a generalization of conventional methods that model analytically derived features such as GVs and MSs explicitly. The reason why is that this algorithm effectively minimizes the divergence without explicit statistical modeling of GVs or MSs. Figure 3.2 illustrates the comparison of the conventional and proposed methods. Also, the discriminator with this algorithm can be interpreted as anti-spoofing, i.e., a technique to detect synthetic speech and prevent voice spoofing attacks. Accordingly, techniques and ideas concerning anti-spoofing can be applied to the acoustic modeling for speech synthesis. In addition, the effects of the divergences are investigated from the perspective of divergence minimization by GANs. This chapter adopts least squares GAN (LS-GAN) and Wasserstein GAN (W-GAN) as image-processing-related

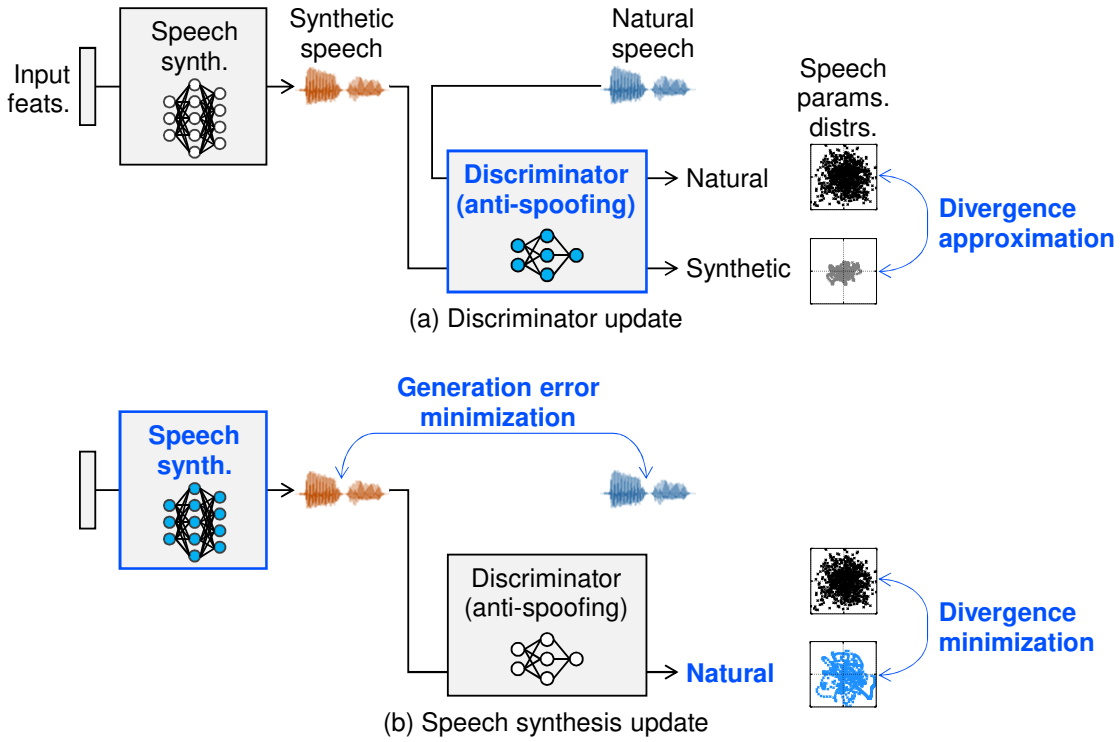

Fig. 3.1. Conceptual diagram of proposed method in Chapter 3

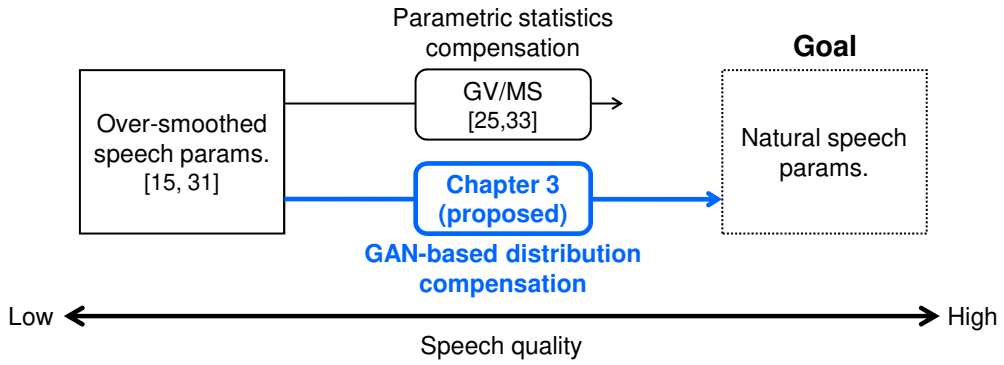

Fig. 3.2. Relation between conventional and proposed methods in Chapter 3

GANs, and f-divergence GAN (f-GAN) as speech-processing-related GANs, to be investigated. This algorithm's effectiveness is evaluated in DNN-based TTS or VC using vocoder parameters.

This chapter is organized as follows (see also Fig. 3.3). Section 3.2 explains a basic framework of GANs. Section 3.3 describes the proposed algorithm for the DNN-based acoustic model training incorporating GANs. Sections 3.4 and 3.5 present experimental evaluations of this algorithm in TTS and VC, respectively. Section 3.6 summarizes this chapter.

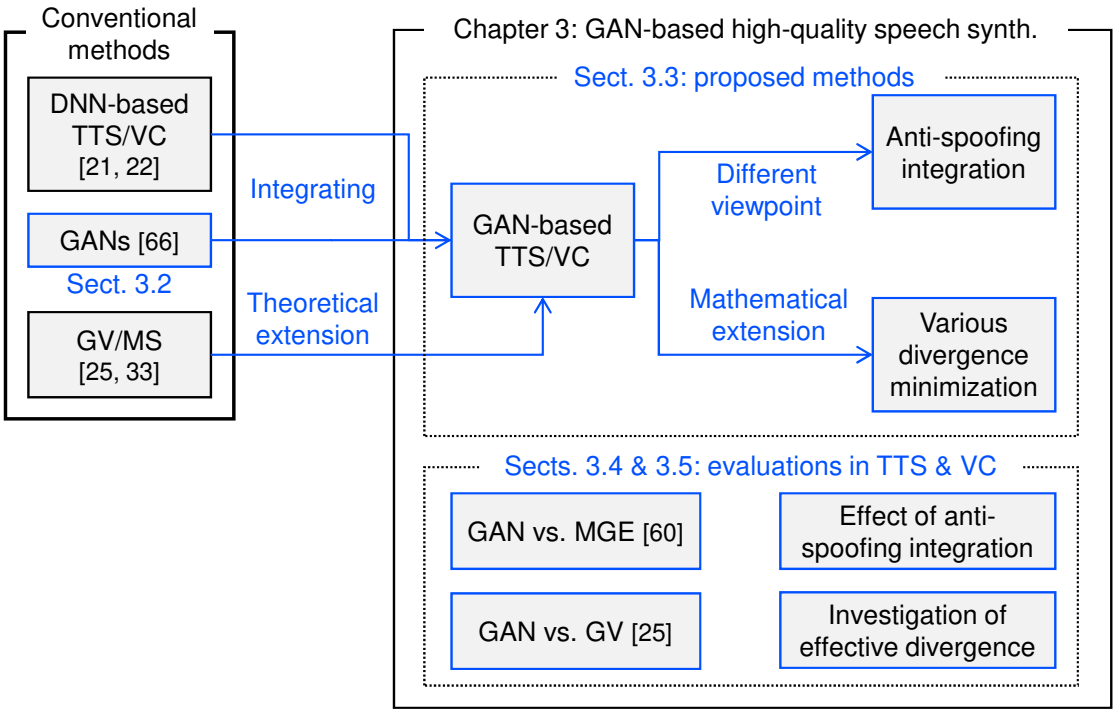

Fig. 3.3. Overview of Chapter 3

3.2 GANs

3.2.1 Objective of GANs

GAN [66] is a class of deep generative models that simultaneously trains two DNNs: a generator and discriminator $D(\boldsymbol{y};\theta_{\mathrm{D}})$. The discriminator's model parameters θ_{D} are 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 $\boldsymbol{y}$ is a natural sample. The discriminator is trained to make the posterior probability 1 for $\boldsymbol{y}$ and 0 for generated $\hat{\boldsymbol{y}}$. Meanwhile, the generator is trained to fool the discriminator, i.e., it tries to make the discriminator make the posterior probability 1 for $\hat{\boldsymbol{y}}$. The two DNNs in the GAN training are iteratively updated by minibatch SGD. Figure 3.4 illustrates a conceptual diagram of the GAN framework.

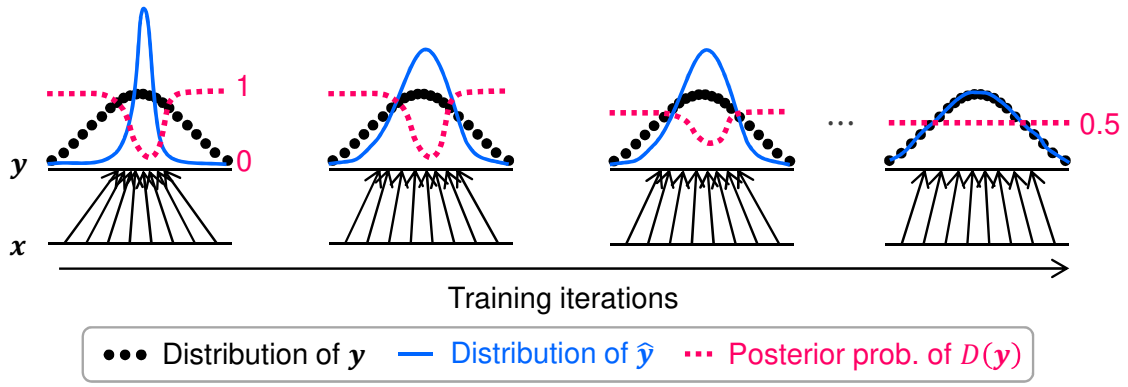

Fig. 3.4. GAN framework. Discriminator is trained to distinguish $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$, while generator is trained to fool it. Here, $\hat{\boldsymbol{y}}$ is generated from $\boldsymbol{x}$ through generator.

3.2.2 Discriminator Training

The discriminator is trained to distinguish $\boldsymbol{y}$ from $\hat{\boldsymbol{y}}$ by minimizing the discriminator loss $L_{\mathrm{D}}^{(\mathrm{GAN})}(\boldsymbol{y}, \hat{\boldsymbol{y}})$ defined as

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

The stochastic gradient $\nabla_{\theta_{\mathrm{D}}} L_{\mathrm{D}}^{(\mathrm{GAN})}(\boldsymbol{y}, \hat{\boldsymbol{y}})$ is used for updating θ_{D}. Figure 3.5 illustrates the procedure for computing the discriminator loss and its gradient.

3.2.3 Generator Training

The generator is trained to fool the updated discriminator by minimizing the adversarial loss $L_{\mathrm{ADV}}^{(\mathrm{GAN})}(\hat{\boldsymbol{y}})$ defined as

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

The stochastic gradient $\nabla_{\theta_{\mathrm{G}}} L_{\mathrm{ADV}}^{(\mathrm{GAN})}(\hat{\boldsymbol{y}})$ is used for updating the generator's model parameters θ_{G}. Goodfellow et al. [66] showed this adversarial framework minimizes the approximated Jensen–Shannon (JS) divergence between distributions of $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$.

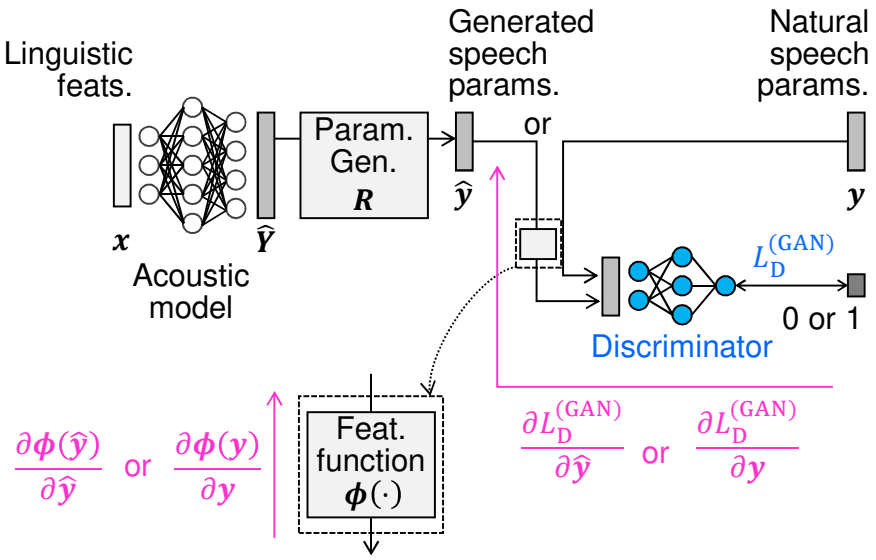

Fig. 3.5. Loss function and gradients for updating discriminator. "Param. Gen." indicates speech parameter generation using MLPG algorithm. Acoustic model's parameters are not updated in this step.

3.3 Acoustic Model Training Using GANs

This section describes a novel training algorithm for a DNN-based acoustic model incorporating GANs.

3.3.1 Acoustic Model Training Criteria Incorporating GANs

The acoustic model with the proposed algorithm is trained to fool the discriminator that distinguishes natural and generated speech parameters. The loss function of the acoustic model training is defined as follows:

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

The second term $L_{\mathrm{ADV}}^{(\mathrm{GAN})}\left(\hat{\boldsymbol{y}}\right)$ is the GAN-derived adversarial loss to make the discriminator recognize the generated speech parameters as natural and minimize the divergence between the distributions of natural and generated speech parameters. Therefore, the proposed algorithm not only minimizes the generation error but also makes the distribution of generated speech parameters close to that of natural ones. The $E_{L_{\mathrm{MGE}}}$ 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

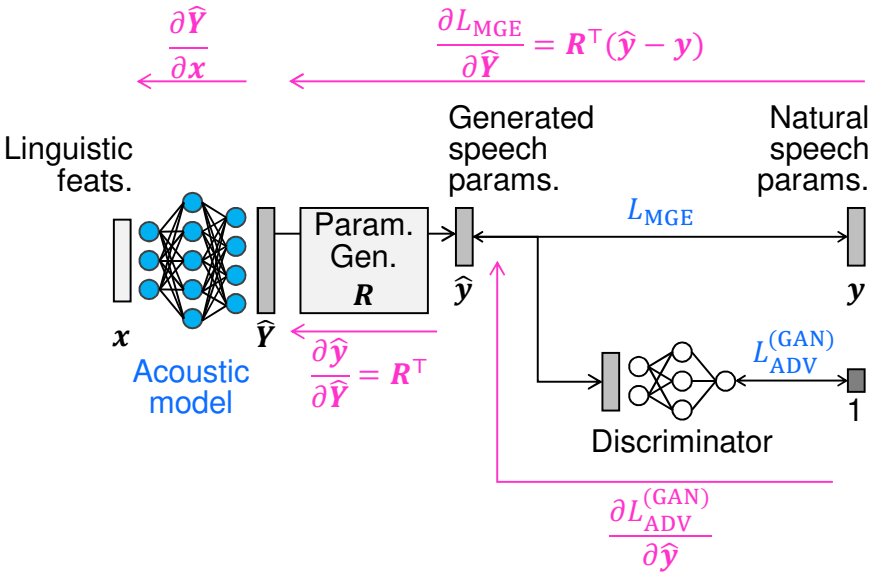

Fig. 3.6. Loss functions and gradients for updating acoustic model in proposed GAN-based algorithm. Model parameters of discriminator are not updated in this step.

$E_{L_{\mathrm{MGE}}}/E_{L_{\mathrm{ADV}}}$ is the scale normalization term between the two loss functions. The hyper-parameter ω_{D} controls the weight of the second term. When $\omega_{\mathrm{D}} = 0$, the loss function is equivalent to the conventional MGE described in Section 2.3.7, and when $\omega_{\mathrm{D}} = 1$, the two loss functions have equal weights. The acoustic model parameters θ_{G} are updated using the stochastic gradient $\nabla_{\theta_{\mathrm{G}}} L_{\mathrm{G}}(\boldsymbol{y}, \hat{\boldsymbol{y}})$. Figure 3.6 illustrates the procedure for computing the proposed loss function and its gradient.

Algorithm 3.1 describes the iterative optimization for the acoustic model and discriminator in the proposed algorithm. When one module is being updated, the model parameters of the other are fixed. For example, the discriminator's model parameters θ_{D} are not updated by the BP algorithm for the acoustic model.

3.3.2 Integrating Anti-spoofing Techniques

The discriminator with the proposed algorithm can be regarded as DNN-based anti-spoofing [29, 67] that distinguishes natural speech from synthetic speech. From this perspective, a feature function $\boldsymbol{\phi}(\cdot)$ can be applied to natural or generated speech parameters as shown in Fig. 3.5. The function calculates more distinguishable features in anti-spoofing than the direct use of speech parameters themselves; i.e., $\boldsymbol{\phi}(\boldsymbol{y})$ and $\boldsymbol{\phi}(\hat{\boldsymbol{y}})$ are used instead of $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$ in Eqs. (3.1) and (3.2), respectively. The stochastic gradient $\partial \boldsymbol{\phi}(\hat{\boldsymbol{y}})/\partial \hat{\boldsymbol{y}}$ is used for the BP algorithm in the acoustic model training.

Algorithm 3.1 Iterative optimization for acoustic model 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 θ_{D} while fixing θ_{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 θ_{G} while fixing θ_{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**

The dynamic features of spectral parameters can be used as the effective features to detect synthetic speech [68]. 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. One can use other features to be incorporated into the proposed algorithm. Because the vocoder systems are based on a minimum-phase vocal tract model, the differences between phase spectra of natural and synthetic speech can be used [69]. Features related to F_0 statistics are also effective to detect spoofing attacks [70, 71] owing to the difficulty in reliable prosody modeling. Temporal magnitude/phase modulation features can be used [72] to capture long-term dependencies of speech parameters.

3.3.3 Duration Model Training Considering Isochrony

The proposed algorithm can be applied to the spectral parameters and continuous F_0 generation straightforwardly. One can also apply the proposed algorithm to phoneme duration generation directly. However, naturally-distributed phoneme durations do not guarantee to have natural isochrony of the target language (e.g., moras in Japanese) [73]. Therefore, the proposed algorithm for duration generation can be modified so that the generated durations naturally distribute in the language-dependent isochrony level. Figure 3.7 shows the architecture to calculate the isochrony-level durations from the phoneme durations. In the case of Japanese, which has mora isochrony, each mora duration is calculated from the corresponding phoneme durations. The discriminator is trained to approximate the di-

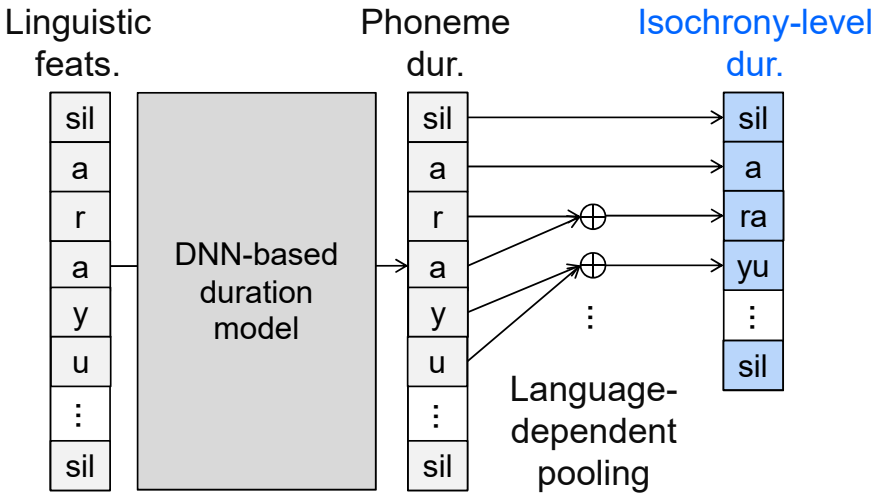

Fig. 3.7. Architecture to calculate isochrony-level duration from phoneme duration. In case of Japanese, which has mora isochrony, each mora duration is calculated from corresponding phoneme durations. For example, mora duration of /ra/ is calculated as sum of phoneme durations of /r/ and /a/.

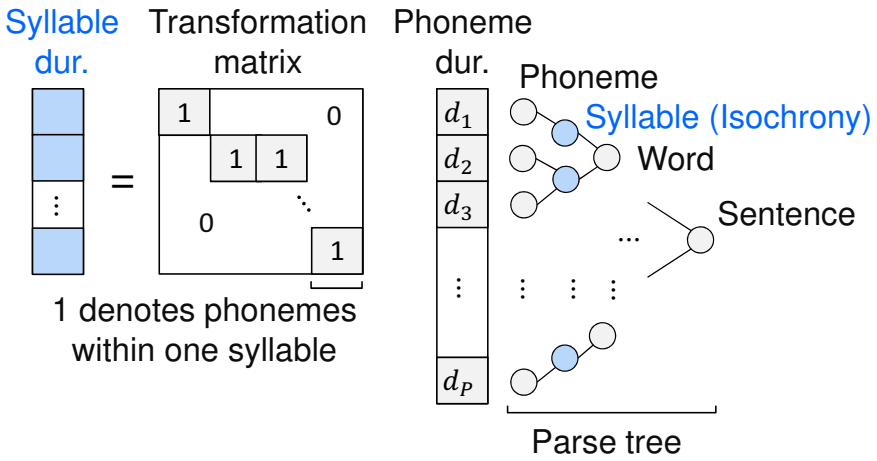

Fig. 3.8. Matrix representation to calculate isochrony-level durations. This is example in syllable-timed language case such as Chinese.

vergence between natural and predicted durations considering the isochrony. Meanwhile, the duration model is trained to minimizes the weighted sum of the MSE between natural and predicted 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. 3.8, the BP algorithm can be done using the transpose of the transformation matrix.

3.3.4 Various Divergences Minimized by GANs

The GAN framework works as a divergence minimization between natural and generated speech parameters. As described in Section 3.2.1, the original GAN [66] minimizes the approximated JS divergence. From the perspective of the divergence minimization, this section discusses additional GANs minimizing other divergences: f-GAN [74], W-GAN [75], and LS-GAN [76]. The divergences encompassed by f-GAN are strongly related to speech processing techniques such as nonnegative matrix factorization [77, 78]. The effectiveness of W-GAN and LS-GAN in the image processing is well 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

f-GAN [74] is the unified framework that encompasses the original GAN [66]. The difference between distributions of natural and generated data is defined as the f-divergence [79], a large class of different divergences including the KL and JS divergences. 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. The $f(\cdot)$ is a convex function satisfying $f(1) = 0$. Although various choices of $f(\cdot)$ for recovering popular divergences are available, this section adopts speech-processing-related ones.

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

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

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

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

Reversed KL (RKL)-GAN: Since the KL divergence is asymmetric, the reversed version, called the RKL divergence $\mathcal{D}_{\mathrm{RKL}}(\boldsymbol{y}\|\hat{\boldsymbol{y}})$ differs from $\mathcal{D}_{\mathrm{KL}}(\boldsymbol{y}\|\hat{\boldsymbol{y}})$ as follows:

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

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

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

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

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

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:

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

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

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

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

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

Note that, the approximated JS divergence minimized by the original GAN is $2\mathcal{D}_{\mathrm{JS}}(\boldsymbol{y}\|\hat{\boldsymbol{y}}) - \log(4)$ [66].

W-GAN

Arjovsky et al. [75] proposed W-GAN that minimizes the earth-mover's distance (Wasserstein-1) to stabilize unstable training of the original GAN. 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 [80], the discriminator loss $L_{\mathrm{D}}^{(\text{W-GAN})}\left(\boldsymbol{y}, \hat{\boldsymbol{y}}\right)$ is defined as follows:

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

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

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

The discriminator is assumed to be the K-Lipschitz function. Therefore, the discriminator's weight parameters are clamped to a fixed interval such as $[-0.01, 0.01]$ after updating the discriminator.

LS-GAN

Mao et al. [76] proposed LS-GAN that formulates the objective function based on the MSE. The MSE-based objective function can avoid the gradient vanishing problem of the original GAN that uses the sigmoid cross entropy. The discriminator loss $L_{\mathrm{D}}^{(\text{LS-GAN})}\left(\boldsymbol{y}, \hat{\boldsymbol{y}}\right)$ is defined as follows:

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

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

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

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, respectively. When these labels satisfy the conditions $b - c = 1$ and $b - a = 2$, the divergence to be minimized corresponds to the Pearson $\mathcal{X}^2$ divergence between $p(\boldsymbol{y}) + q(\hat{\boldsymbol{y}})$ and $2q(\hat{\boldsymbol{y}})$. Because results of preliminary experiments showed that these conditions degraded synthetic speech quality, alternative conditions suggested in Eq. (9) of [76], i.e., $a = 0$, $b = 1$, and $c = 1$, were used in this thesis.

3.3.5 Discussion

The proposed loss function (Eq. (3.3)) is the combination of a multi-task learning algorithm using a discriminator [81] and GANs. In defining $L_{\mathrm{G}}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = L_{\mathrm{ADV}}^{(\mathrm{GAN})}(\hat{\boldsymbol{y}})$, the loss function is equivalent to that for GANs. The proposed algorithm considers a fully supervised setting unlike GANs, i.e., it utilizes the referred input and output parameters [82] without latent variables for the training. Since only the BP algorithm is used for training, a variety of DNN architectures such as LSTM [83] can be used for the acoustic model and discriminator.

The use of the feature function $\boldsymbol{\phi}(\cdot)$ for the proposed algorithm enables choosing not only analytically derived features (e.g., GVs and MSs) but also automatically derived features (e.g., auto-encoded features [84]) to be reproduced. However, the features that are effective in anti-spoofing may degrade synthetic speech quality because they do not always relate to human speech perception.

The proposed algorithm makes the distribution of generated speech parameters close to that of natural ones. Figure 3.9 plots natural and generated speech parameters with several mel-cepstral coefficient pairs to illustrate this effect clearly. The proposed algorithm widens the distributions of generated speech parameters as much as those of natural ones, whereas "MGE" does not. This effect is strongly observed in the distribution of higher-order of mel-cepstral coefficients. A noteworthy fact is that the proposed algorithm compensates for the distribution differences between natural and generated speech parameters better than "MGE-GV," which can be expected to improve synthetic speech quality.

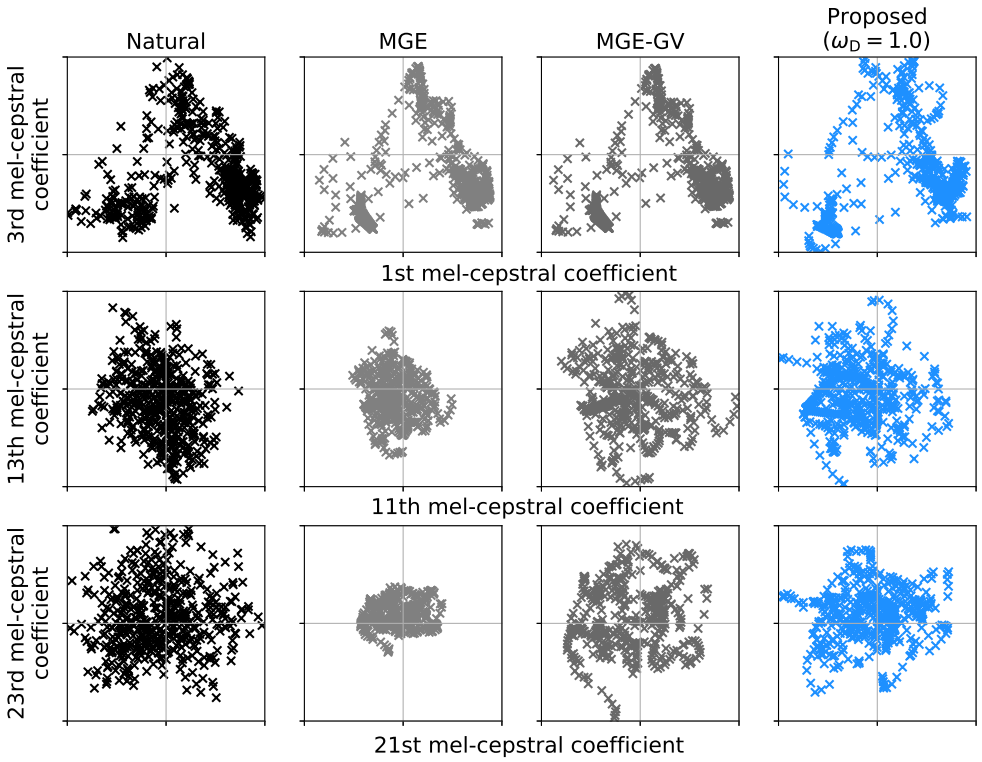

Fig. 3.9. Scatter plots of mel-cepstral coefficients with several pairs of orders. From left, figures correspond to natural speech, conventional MGE algorithm, conventional MGE algorithm with GV compensation, and proposed algorithm ($\omega_D = 1.0$), respectively. These mel-cepstral coefficients were extracted from one utterance of evaluation data.

One can explore what components (e.g., analytically derived features and intuitive reasons [85]) the proposed algorithm changes. Firstly, GVs of natural and generated mel-cepstral coefficients were calculated. Figure 3.10 shows the calculation results. The proposed algorithm reduced the GV difference between natural and generated speech parameters. This is a quite natural result because compensating for the distribution differences is related to minimizing the moment differences [86, 87]. Secondly, a maximal information coefficient (MIC) [88] of speech parameters was calculated to quantify a nonlinear correlation among them. Figure 3.11 shows the calculation results. Weak correlations are observed among the natural speech parameters as reported in [89]. On the other hand, strong correlations are observed among speech parameters generated by the conventional MGE algorithm. The proposed algorithm weakens the correlations among generated speech parameters. These results indicate that the proposed algorithm reproduces not only natural GVs but also natural correlation among speech parameters. Finally, the statistics of continuous F_0, phoneme duration, and mora duration were calcu-

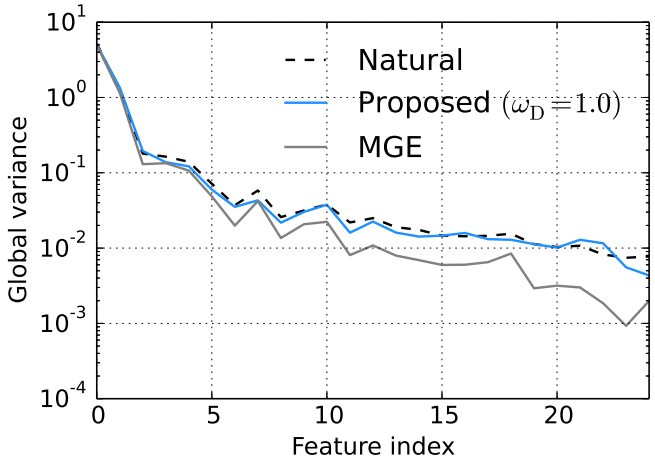

Fig. 3.10. Averaged GVs of natural and generated mel-cepstral coefficients. Black dashed line denotes GVs of natural mel-cepstral coefficients. Gray and blue lines correspond to GVs of mel-cepstral coefficients generated by conventional MGE and proposed algorithms, respectively.

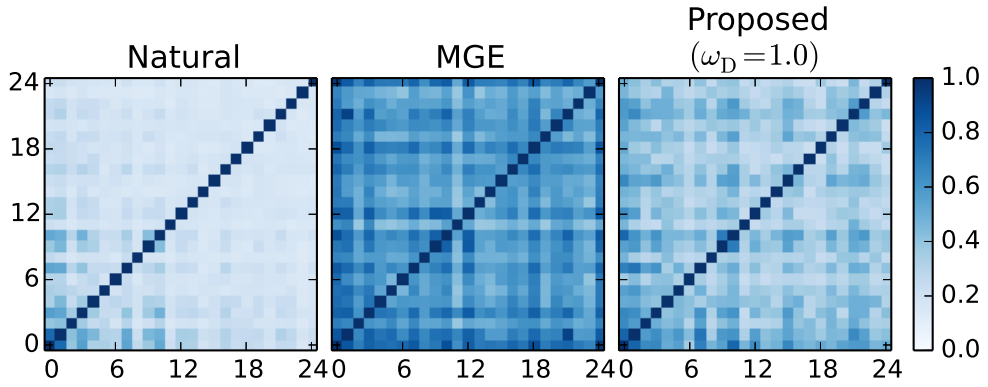

Fig. 3.11. MICs of natural and generated mel-cepstral coefficients. MIC ranges from 0.0 to 1.0, and two variables with strong correlation have value closer to 1.0. From left, figures correspond to natural speech parameters, generated ones by conventional MGE algorithm, and generated ones by proposed algorithm, respectively. These MICs were calculated from one utterance of evaluation data.

lated. Tables 3.1, 3.2, and 3.3 list the calculation results regarding continuous F_0, phoneme duration, and mora duration, respectively. The bold values in these tables are the closest to natural statistics in the results. In Tables 3.2 and 3.3, "Proposed (phoneme)" and "Proposed (mora)" are the proposed algorithms using phoneme and mora durations to calculate the adversarial loss, respectively. These tables indicate that the proposed algorithm also makes the statistics closer to those of natural speech than the conventional algorithm. In the results concerning duration generations, "Proposed (mora)" tends to

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**

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, $\omega_{\mathrm{D}} = 1.0$)	14.963	**75.471**
Proposed (mora, $\omega_{\mathrm{D}} = 1.0$)	**15.074**	73.207

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

reduce the mean difference rather than the variance difference.

The proposed algorithm for spectrum and F_0 generation (Section 3.2.3) learns the joint distribution of them. Therefore, one can perform the distribution compensation considering correlations [90] between different features. Also, the dimensionality differences [91] can be applied to fool the discriminator. Since the time resolutions in a phoneme duration and mora duration are different, the proposed algorithm considering the language-dependent isochrony is related to multi-resolution GAN [92] and hierarchical duration modeling [93].

Regarding related work, Kaneko et al. [94] proposed a GAN-based post-filter for TTS. The post-filtering has high portability because it is independent of original speech synthesis procedures. However, it comes at a high computational cost and has a heavy disk footprint

Table 3.4. Architectures of DNNs used in TTS evaluations. Feed-Forward DNNs 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 model	274–400×3–75	442–512×3–94	442–512×3–94
Discriminator	25–200×2–1	26–256×3–1	1–256×3–1
Duration model	N/A	439–256×3–1	439–256×3–1

in the speech synthesis phase. In contrast, the proposed algorithm does not require additional resources (i.e., computational cost and other modules) while achieving high-quality speech synthesis.

3.4 Experimental Evaluations for TTS

3.4.1 Conditions for TTS Evaluation

A speech corpus of a male speaker was used. The speaker uttered 503 phonetically balanced sentences [95]. The numbers of sentences for the training and evaluation were 450 (subsets A to I) and 53 (subset J), respectively. Speech signals were sampled at a rate of 16 kHz. The shift length was set to 5 ms. The 0th-through-24th mel-cepstral coefficients were used as spectral parameters. F_0 and 5 band-aperiodicity [44, 96] were used as excitation parameters. The STRAIGHT vocoder [47] was used for speech parameter extraction and speech waveform synthesis. Speech parameter trajectory smoothing [97] with a 50 Hz cutoff modulation frequency was applied to the spectral parameters in the training data for improving training accuracy. Eighty percent of the silent frames were removed from the training data.

Table 3.4 lists the DNN architectures used for the evaluation. The ReLU [54] was used for the activation function for the hidden layers. The linear function was used for the output activation function of the acoustic model and duration model. The sigmoid function was used for the output activation function of the discriminator. In the spectral parameter generation (Sections 3.4.2 through 3.4.4), the acoustic model predicted a 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.). Since F_0, band-aperiodicity, and duration of natural speech were directly used for the speech waveform synthesis, some of the prosody-related features, such as the accent type, were included in the linguistic features. In the spectral parameter and F_0 generation (Section 3.4.5), the acoustic model predicted a static-dynamic feature sequence of the mel-cepstral coefficients, continuous log F_0 [46], 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), the duration model was trained to predict phoneme durations from corresponding linguistic features (439-dim).

In the training phase, speech parameters and real-valued linguistic features were normalized to have zero-mean and unit-variance. The training algorithm based on minimizing the MSE (Eq. (2.15)) [21] was first performed with 25 iterations, then the conventional MGE training algorithm [60] was performed with 25 iterations for the initialization of the acoustic model. Here, "iteration" means using all the training data (450 utterances) once for the training. The discriminator was initialized using natural speech parameters and generated ones after the MGE training. The number of iterations for the discriminator initialization was 5. The proposed algorithm was finally performed with 25 iterations using the initialized acoustic model and discriminator. The expectation values $E_{L_{\mathrm{MGE}}}$ and $E_{L_{\mathrm{ADV}}}$ were estimated at each iteration. The optimization algorithm was AdaGrad [98]. The learning rate was set to 0.01.

3.4.2 Objective Evaluation of Spectral Parameter Generation

The parameter generation loss defined in Eq. (2.16) and spoofing rate were calculated to evaluate the proposed algorithm objectively. 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 generated speech parameter that fools the discriminator. The discriminator for the spoofing rate calculation was trained using natural speech parameters and generated speech parameters of the conventional MGE training. The generation loss and spoofing rate were calculated with various settings of the hyperparameter ω_{D}.

Figure 3.12 shows the evaluation results. The generation loss monotonically increases as ω_{D} increases from 0.0 to 0.4. However, this tendency is not observed when $\omega_{\mathrm{D}} > 0.4$. On the other hand, the spoofing rate significantly increases as ω_{D} increases from 0.0 to 0.2 and becomes almost nearly 1.0 when $\omega_{\mathrm{D}} > 0.3$. These results demonstrate that the

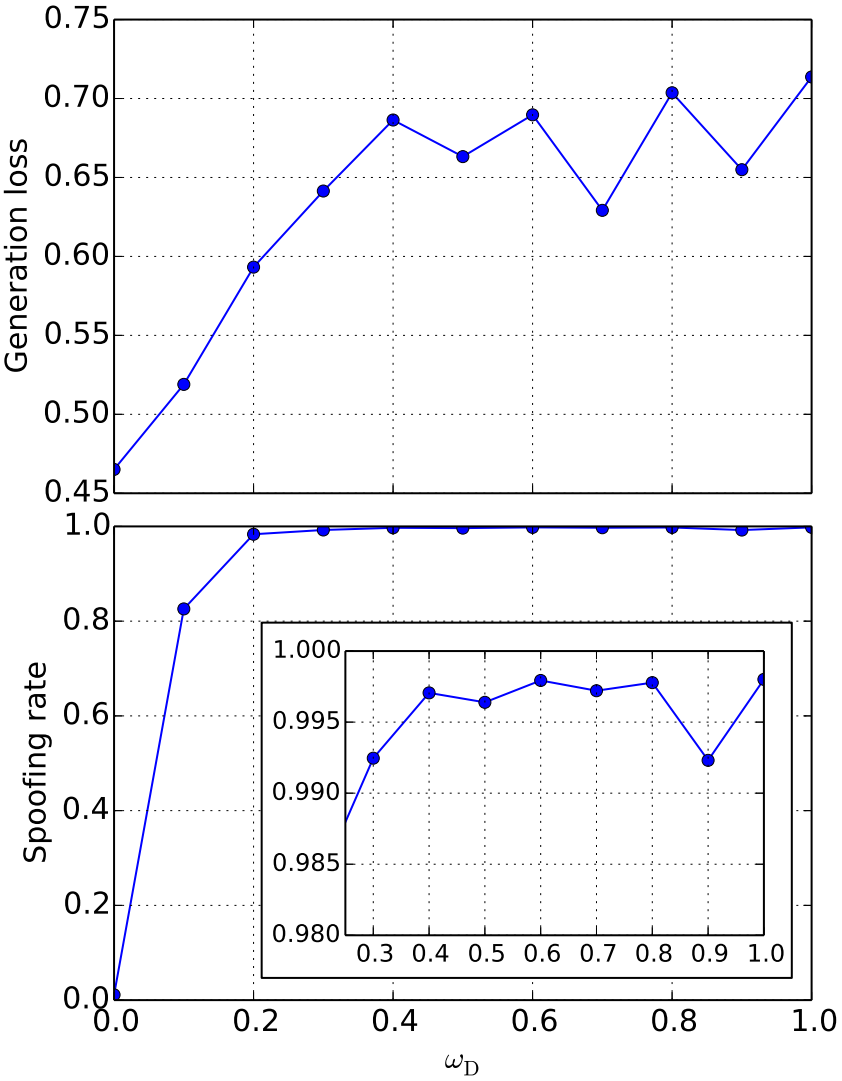

Fig. 3.12. Parameter generation loss (above) and spoofing rate (below) curves against various settings of hyperparameter ω_{D}

proposed algorithm increases the generation loss, but it can train the acoustic model to fool the discriminator. In other words, the proposed algorithm tries to reduce the statistical difference between natural and generated speech parameters, although it does not necessarily decrease the generation error.

3.4.3 Investigation of Convergence

The proposed algorithm was performed with 100 iterations to investigate the convergence property. Figure 3.13 plots the generation loss and adversarial loss curves for the training and evaluation data. This figure indicates that both loss values are almost monotonically

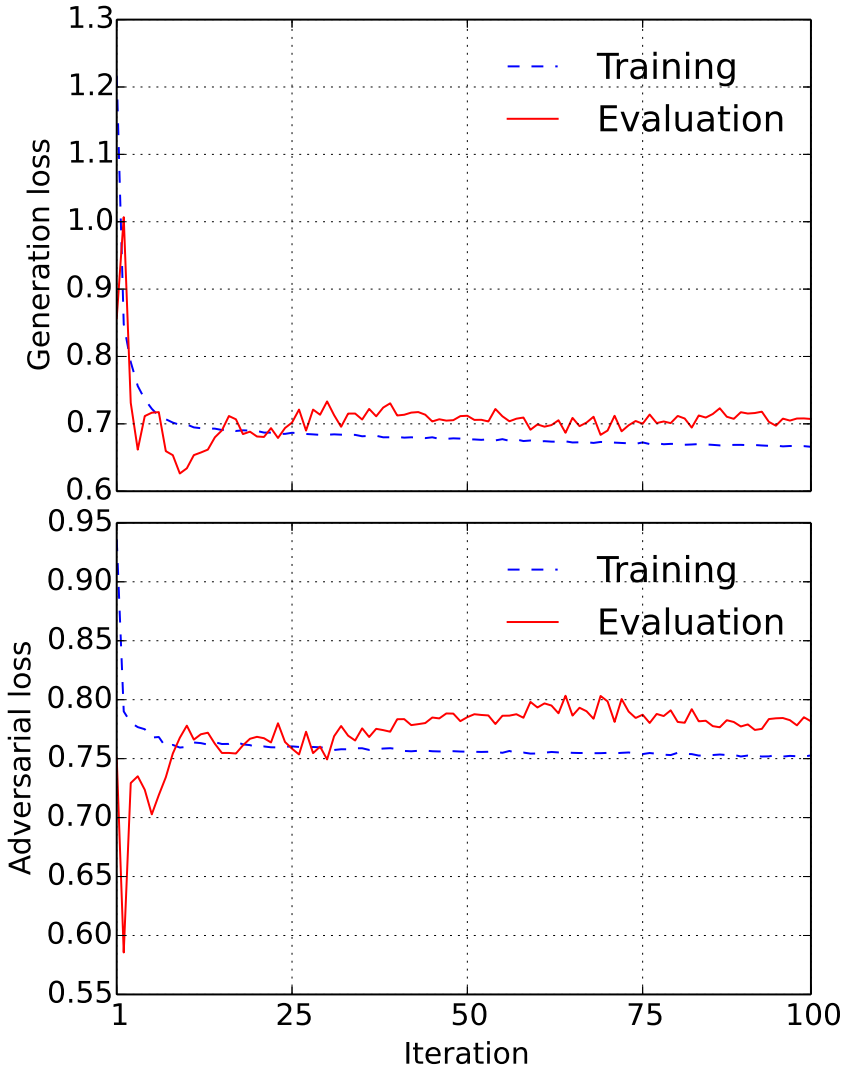

Fig. 3.13. Parameter generation loss (above) and adversarial loss (below) curves for training data (blue-dashed line) and evaluation data (red line)

decreased in training. Although the loss values for the 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 proposed algorithm. Speech samples were generated with three algorithms:

MGE: conventional MGE algorithm (= Proposed ($\omega_{\mathrm{D}} = 0.0$))

Proposed ($\omega_{\mathrm{D}} = 0.3$)**:** proposed algorithm using least hyperparameter setting that achieves spoofing rate > 0.99

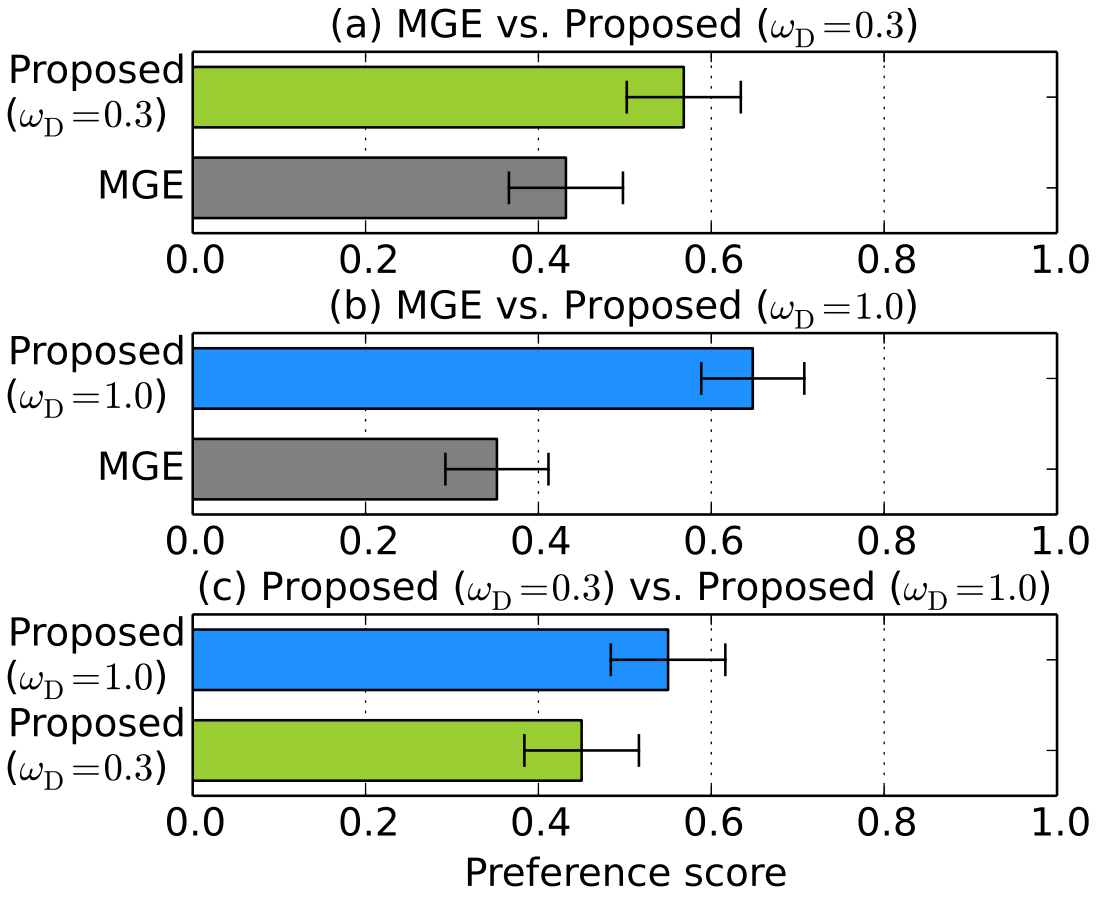

Fig. 3.14. Preference scores of speech quality with 95% confidence intervals (spectral parameter generation in TTS). From top, numbers of listeners were 22, 24, and 22, respectively.

Proposed ($\omega_D = 1.0$)**:** proposed algorithm using standard hyperparameter setting

Pairs of speech samples generated by each algorithm were presented to listeners in random order. Listeners participated in the assessment by using crowdsourced subjective evaluation systems.

Figure 3.14 shows the evaluation results. In Figs. 3.14(a) and (b), the proposed algorithm significantly outperforms the conventional MGE algorithm in both hyperparameter settings. These results demonstrate that the proposed algorithm robustly yields significant improvement in terms of synthetic speech quality regardless of its hyperparameter setting. Henceforth, the hyperparameter ω_D was set to 1.0 for the following evaluations because Fig. 3.14(c) shows that the score of "Proposed ($\omega_D = 1.0$)" was slightly better than that of "Proposed ($\omega_D = 0.3$)."

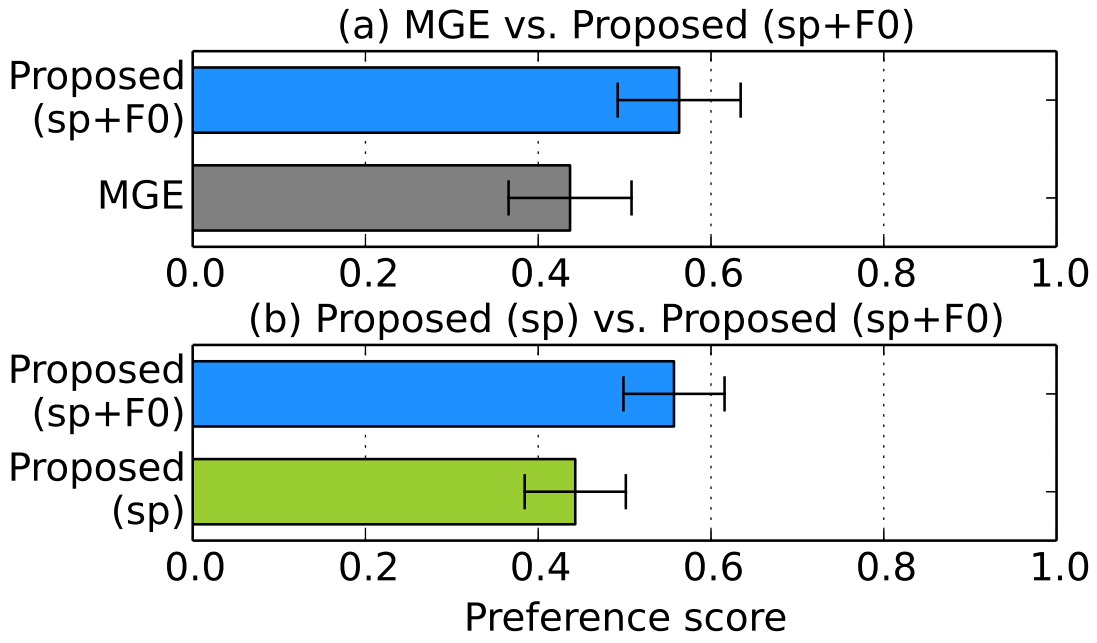

Fig. 3.15. Preference scores of speech quality with 95% confidence intervals (spectral parameter and F_0 generation in TTS). From top, numbers of listeners were 19 and 28, respectively.

3.4.5 Subjective Evaluation of F_0 Generation

The effectiveness of the proposed algorithm for F_0 generation was investigated. Preference AB tests were conducted using the following three algorithms:

MGE: conventional MGE algorithm
Proposed (sp): proposed algorithm applied to only spectral parameters
Proposed (sp+F0): proposed algorithm applied to both spectral parameters and F_0

Pairs of speech samples generated by each algorithm were presented to listeners in random order. Here, "Proposed (sp)" was not compared with "MGE" since Fig. 3.14 has already demonstrated that the proposed algorithm improved synthetic speech quality in spectral parameter generation. Listeners participated in the assessment by using crowdsourced subjective evaluation systems.

Figure 3.15 shows the evaluation results. The score of "Proposed (sp+F0)" is significantly higher than those of "Proposed (sp)" and "MGE." These results demonstrate the effectiveness of the proposed algorithm not only for spectral parameters but also for F_0.

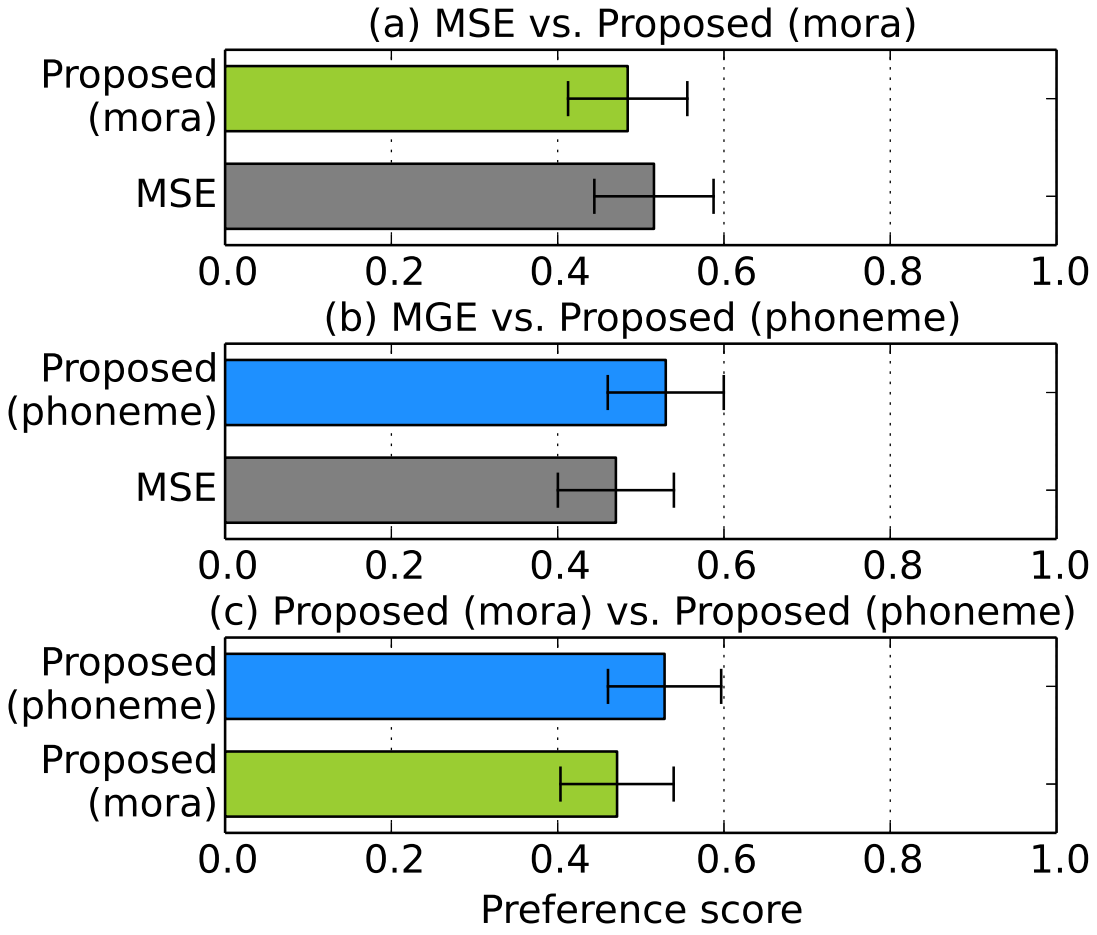

Fig. 3.16. Preference scores of speech quality with 95% confidence intervals (duration generation in TTS). From top, numbers of listeners were 19, 20, and 21, respectively.

3.4.6 Subjective Evaluation of Duration Generation

The effectiveness of the proposed algorithm in duration generation was investigated. Preference AB tests were conducted using the following three algorithms:

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

Figure 3.16 shows the evaluation results. There are no significant differences in the resulting scores. The classification accuracy of DNN-based anti-spoofing was calculated to investigate the reason. The anti-spoofing was trained to distinguish natural speech parameters from generated ones by the conventional MSE or MGE algorithm. The pro-

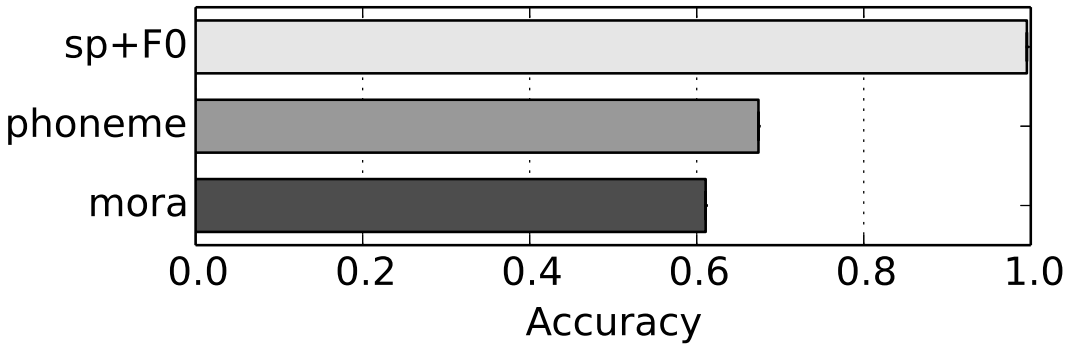

Fig. 3.17. Classification accuracy of anti-spoofing. "sp+F0", "phoneme", and "mora" denote using spectral parameters and F_0, phoneme durations, and mora durations to discriminate natural and synthetic speech, respectively.

posed algorithm can be expected to work effectively when the classification accuracy is higher; i.e., the statistical differences between natural and generated speech parameters are larger. Figure 3.17 shows the classification accuracy. The accuracy of anti-spoofing using phoneme or mora duration is lower than that using spectral parameters and F_0. These results suggest that the distribution compensation by the proposed algorithm does not work well in duration generation. Henceforth, the proposed algorithm was not applied to duration generation.

3.4.7 Comparison to GV Compensation

Figure 3.10 demonstrated that the proposed algorithm reproduced GVs of natural speech parameters. Here, this algorithm was compared with GV compensation to investigate whether the former improves speech quality more than the latter. The GV post-filtering [63] was applied to the spectral parameters and F_0 generated by the MGE training. A preference AB test involving 29 listeners was conducted using crowdsourced subjective evaluation systems.

Figure 3.18 shows the evaluation result. The score of "Proposed" is significantly higher than that of the conventional GV post-filter ("MGE-GV"). This result shows that the proposed algorithm produces more gain in speech quality than GV compensation.

3.4.8 Effect of Feature Function

The effectiveness of the feature function for anti-spoofing was investigated. The following two functions were compared:

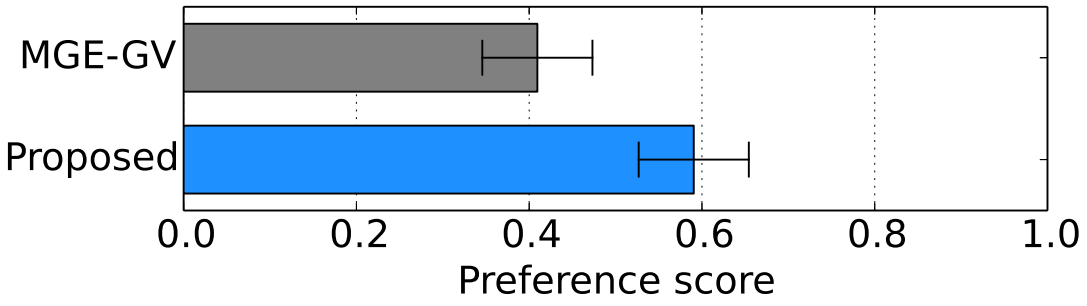

Fig. 3.18. Preference scores of speech quality with 95% confidence intervals (comparison with GV compensation)

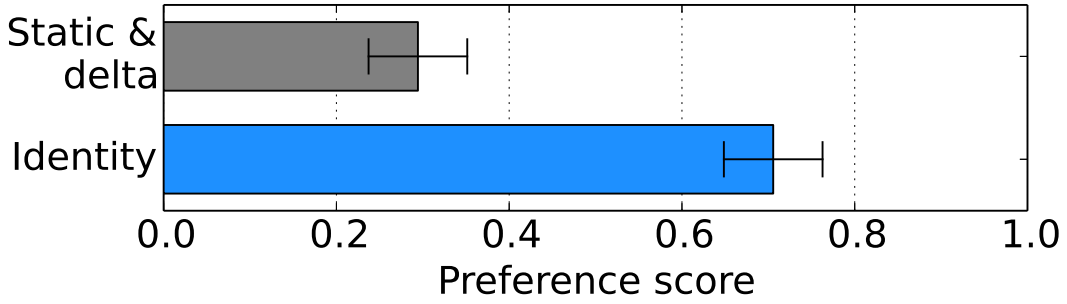

Fig. 3.19. Preference scores of speech quality with 95% confidence intervals (effect of feature function)

Identity: $\boldsymbol{\phi}(\boldsymbol{y}) = \boldsymbol{y}$
Static & delta [68]: $\boldsymbol{\phi}(\boldsymbol{y}) = \boldsymbol{M}\boldsymbol{y}$

"Identity" is equivalent to not using the feature function. When "Static & delta" was adopted, joint vectors of the static, delta, and delta-delta mel-cepstral coefficients and continuous F_0 were used for the discriminator. A preference AB test involving 31 listeners was conducted using crowdsourced subjective evaluation systems.

Figure 3.19 shows the evaluation result. The score of "Static & delta" is much lower than that of "Identity." This result suggests that "Static & delta" effectively distinguishes natural and synthetic speech, but does not improve speech quality.

3.4.9 Subjective Evaluation Using Richer DNN Architecture

Only simple Feed-Forward DNNs were used in the above-described evaluations. Here, two-layer uni-directional LSTM [83] was used for both the acoustic model and discriminator to investigate the effectiveness of using a richer DNN architecture. The numbers of memory

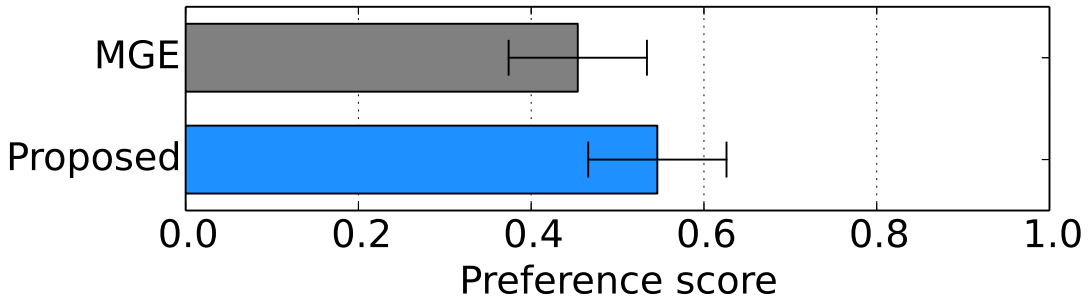

Fig. 3.20. Preference scores of speech quality with 95% confidence intervals (effect of LSTM-based acoustic model and discriminator)

cells in the acoustic model and discriminator were 256 and 128, respectively. The proposed algorithm was applied to spectral parameters and F_0. The MGE ("MGE") and proposed ("Proposed") algorithms were compared. A preference AB test involving 19 listeners was conducted using crowdsourced subjective evaluation systems.

Figure 3.20 shows the evaluation result. The score of "Proposed" is higher than that of "MGE," demonstrating that the proposed algorithm works for not only simple DNN architectures but also richer ones.

3.4.10 Effect of Divergence to Be Minimized by GANs

As the final investigation regarding TTS, various GANs were adopted to the proposed algorithm. The following GANs were compared:

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)

A five-scale point mean opinion score (MOS) test on speech quality was conducted. Speech samples generated with each GAN were presented to listeners in random order. Fifty-five listeners participated in the assessment by using crowdsourced subjective evaluation systems.

Figure 3.21 shows the evaluation results. The proposed algorithm works in the cases of all GANs except for "KL-GAN" and "JS-GAN." In addition, two points are noteworthy:

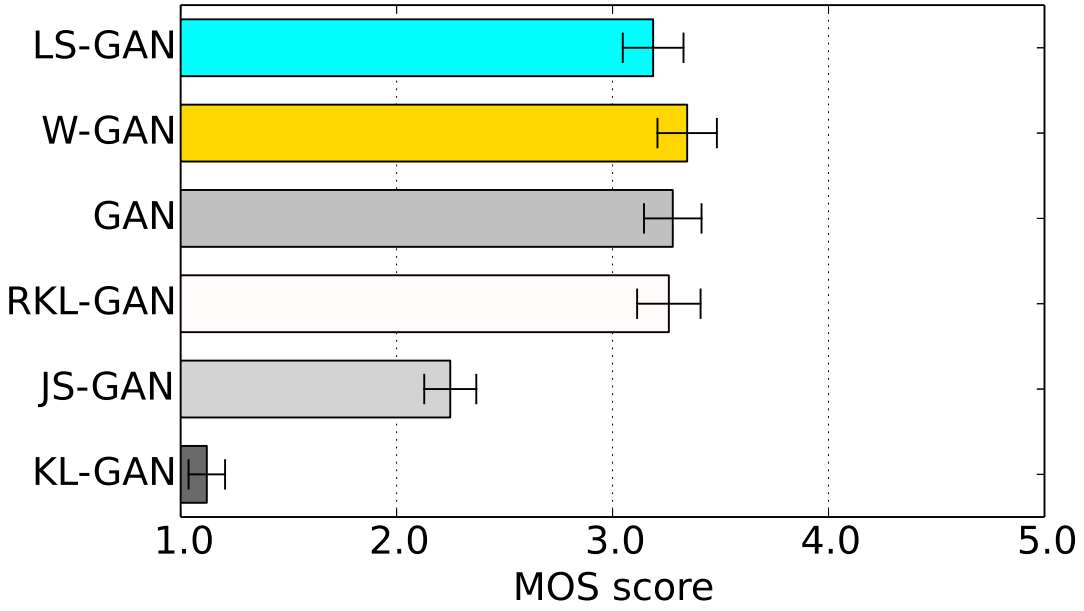

Fig. 3.21. MOS evaluation results of speech quality with 95% confidence intervals (comparing various GANs)

1) minimizing the KL-divergence ("KL-GAN") does not improve synthetic speech quality, but the reversed version ("RKL-GAN") works, and 2) the JS-divergence ("JS-GAN") does not work well, but the approximated version ("GAN") works. One possible reason is the behavior of each GAN in density fitting: mode-seeking or mode-covering. As reported in [99], when the target distribution is multi-modal, "RKL-GAN" and "GAN" tend to result in mode-seeking learning while "KL-GAN" and "JS-GAN" lead to mode-covering learning. These insights indicate that the mode-seeking-based learning is suitable for the acoustic model training in speech synthesis. The best GAN in terms of synthetic speech quality is W-GAN, whose MOS value is significantly higher than those of LS-GAN, JS-GAN, and KL-GAN. This result indicates that the stable optimization in GAN-based acoustic model training is crucial for improving synthetic speech quality.

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 of VC using speech parameter conversion, two DNNs were trained: one for male-to-male VC and the other for male-to-female VC. Architectures for the acoustic model and discriminator were Feed-Forward DNNs. The hidden layers of the acoustic model and discriminator had 512×3 units and 256×3 units, respectively. The 1st-through-59th mel-cepstral coefficients were converted by DNNs. 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 frame lengths of the input and output speech parameters.

In evaluation of VC using spectral differentials, DNNs for male-to-male VC were trained. Here, input-to-output highway networks (described in Appendix A) were adopted to the acoustic model instead of Feed-Forward DNNs. The transform gate of the highway networks only had a 59-unit input and 59-unit sigmoid output layers.

Speech samples were generated with the conventional MGE and proposed algorithms. Preference AB tests were conducted to evaluate converted speech quality. Pairs of speech samples of the two sets were presented to listeners in random order. Listeners selected speech samples that sounded better in quality. Similarly, XAB tests on the speaker individuality were conducted using natural speech of the target speaker as the reference "X."

3.5.2 Subjective Evaluation Using Speech Parameter Conversion

Eight listeners participated in assessment of male-to-male VC case, and 27 listeners participated in assessment of male-to-female VC case by using crowdsourced subjective evaluation systems. Figures 3.22 and 3.23 show the evaluation results regarding the speech quality and speaker individuality, respectively. The results shown in Fig. 3.22 demonstrate that the proposed algorithm significantly improves the speech quality. Similar tendency is observed in the evaluation results regarding the speaker individuality shown in Fig. 3.23. These improvements are observed in both male-to-male and male-to-female VC settings. One of the reasons for these improvements are GV reproduction by the proposed algorithm, which affects not only speech quality but also speaker individuality [25].

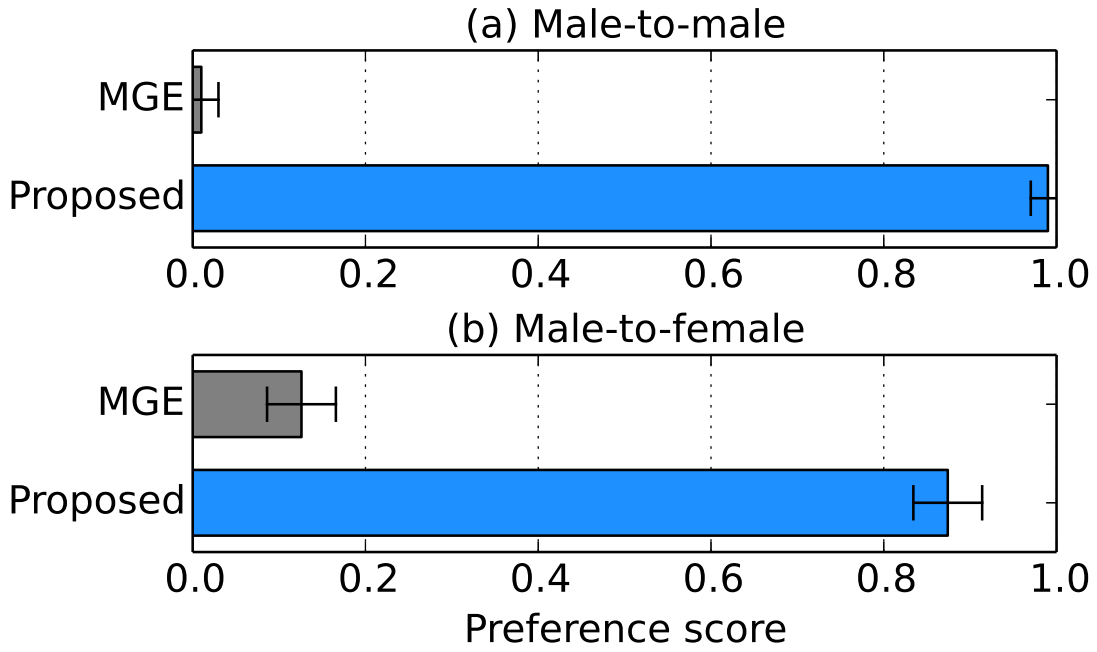

Fig. 3.22. Preference scores of speech quality with 95% confidence intervals (DNN-based VC using speech parameter conversion)

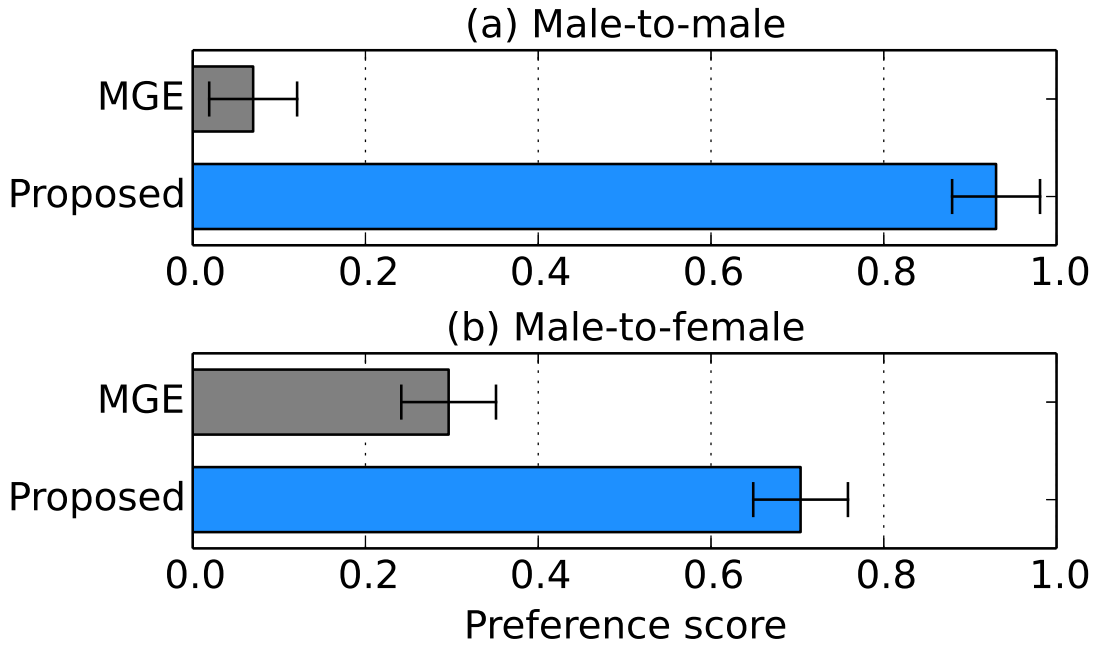

Fig. 3.23. Preference scores of speaker individuality with 95% confidence intervals (DNN-based VC using speech parameter conversion)

3.5.3 Subjective Evaluation Using Spectral Differentials

The number of listeners that participated in each of the preference AB or XAB tests was eight. Figures 3.24 and 3.25 show the evaluation results regarding the speech quality and

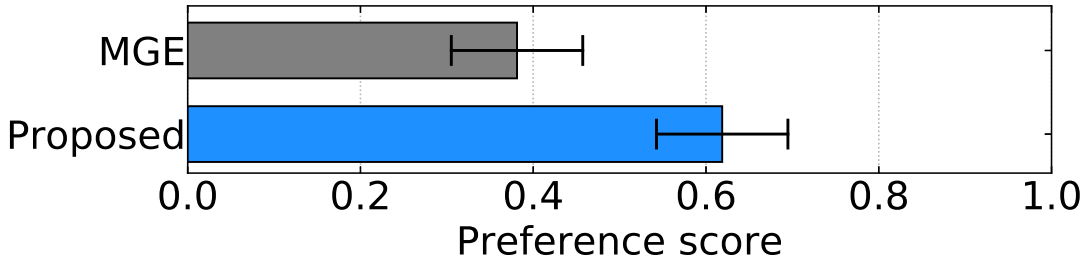

Fig. 3.24. Preference scores of speech quality with 95% confidence intervals (DNN-based VC using spectral differentials)

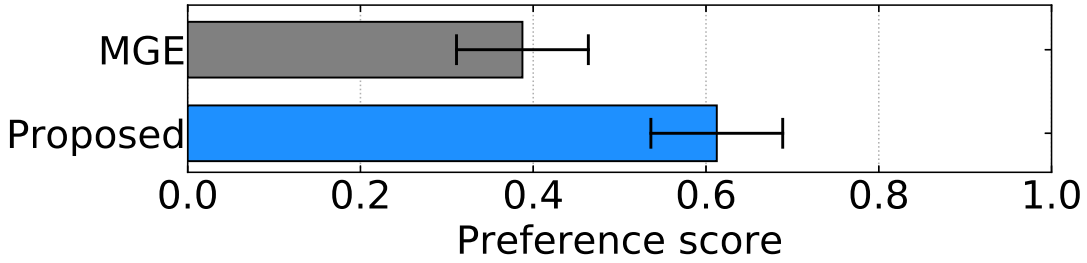

Fig. 3.25. Preference scores of speaker individuality with 95% confidence intervals (DNN-based VC using spectral differentials)

speaker individuality, respectively. The results demonstrate that the proposed algorithm is also effective in VC using spectral differentials.

3.5.4 Evaluation in Many-to-one VC

Experimental evaluation of the proposed algorithm was conducted in many-to-one VC. The details of the evaluation are described in Appendix B. In summary, the proposed algorithm is effective to improve converted speech quality in not only one-to-one VC but also many-to-one VC.

3.6 Summary

This chapter proposed a novel training algorithm for DNN-based high-quality speech synthesis. This algorithm incorporates a framework of GANs that adversarially trains a generator and discriminator to learn high-fidelity deep generative models. The acoustic model with this algorithm is trained to fool the discriminator that distinguishes natural and synthetic speech. Since the GAN framework minimizes the divergence between natural and generated data distributions, the acoustic model can be trained to not only minimize

the generation loss but also make the distribution of generated speech parameters close to that of natural ones. As a result, this algorithm reproduces not only natural GVs but also natural correlation among speech parameters. Experimental evaluations were conducted in DNN-based TTS or VC using vocoder parameters. The evaluation results indicated that this algorithm yielded significant improvements in terms of speech quality regardless of its hyperparameter settings. The results also showed that the use of W-GAN in this algorithm improved synthetic speech quality the best in comparison with various GANs.

Chapter 4

Vocoder-free Statistical Speech Synthesis Using GANs

4.1 Introduction

This chapter extends the GAN-based acoustic model training algorithm proposed in Chapter 3 to a vocoder-free speech synthesis method using STFT spectra. The vocoder-free method can incorporate other DNN-based signal-processing techniques applied in the time-frequency domain, such as speech enhancement [100] and speech separation [101], into the acoustic model training. However, it is difficult to apply the GAN-based algorithm to the vocoder-free method directly. One of the reasons is that the distribution of high-dimensional STFT amplitude spectra (e.g., 1,024-dim.) is more complicated than that of the low-dimensional vocoder parameters. This chapter proposes a novel algorithm based on GANs using *low-frequency-resolution amplitude spectra* to train a DNN-based acoustic model. Figure 4.1 shows the conceptual diagram of this algorithm. The frequency resolution of amplitude spectra is first lowered using an average-pooling function along with a frequency axis while keeping rough structures of the spectra (i.e., the spectral envelope) preserved. The GAN-based training algorithm is then performed using the low-frequency-resolution amplitude spectra to calculate the adversarial loss. Since the spectral envelopes are dominant features in perceiving the quality of synthetic speech, this algorithm can be expected to improve synthetic speech quality better than using the GANs in the original frequency resolution. Furthermore, a frequency warping function can be applied to amplitude spectra for introducing various frequency scales that are related to human speech perception (e.g., mel and inverse mel frequency scales). This chapter also proposes a training algorithm based on GANs using *multi-frequency-resolution am-*

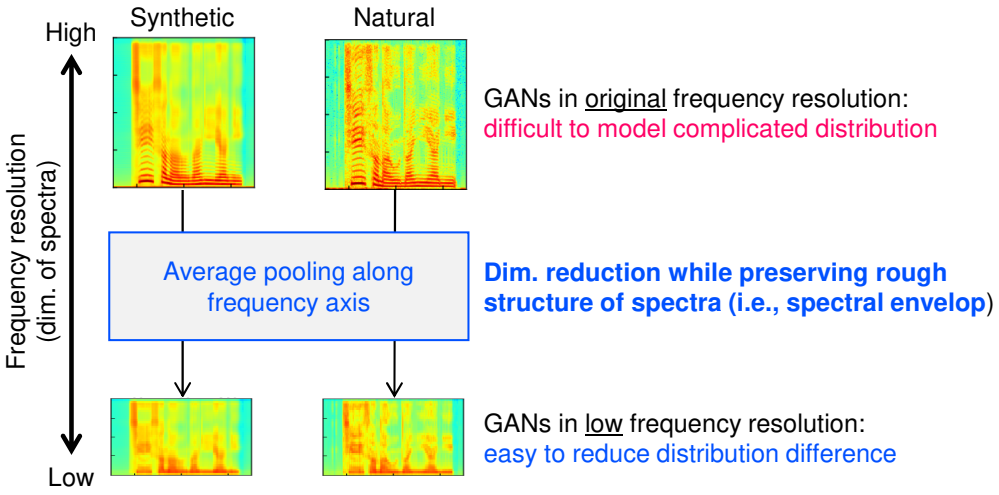

Fig. 4.1. Conceptual diagram of proposed method in Chapter 4

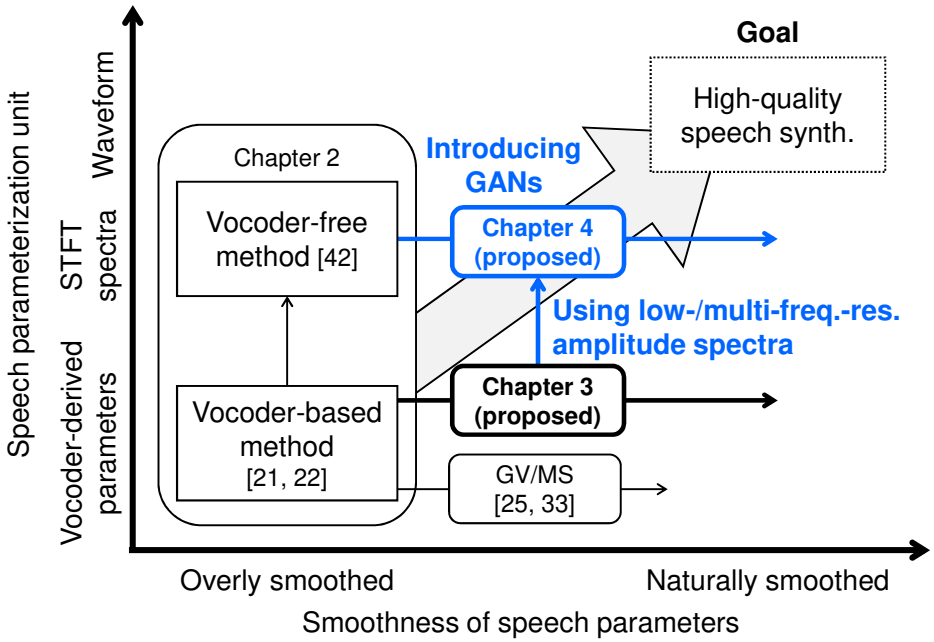

Fig. 4.2. Relation between conventional and proposed methods in Chapter 4

plitude spectra that uses both low- and original-frequency-resolution amplitude spectra. This algorithm can be expected to compensate for not only the differences in *rough* structures but also in *fine* structures of natural and generated amplitude spectra. Figure 4.2 illustrates the relation between conventional and proposed methods.

This chapter is organized as follows (see also Fig. 4.3). Section 4.2 describes the proposed GAN-based algorithms using low- and multi-frequency-resolution amplitude spectra. Section 4.3 presents experimental evaluations of these algorithms in DNN-based vocoder-free TTS using STFT spectra. Section 4.4 summarizes this chapter.

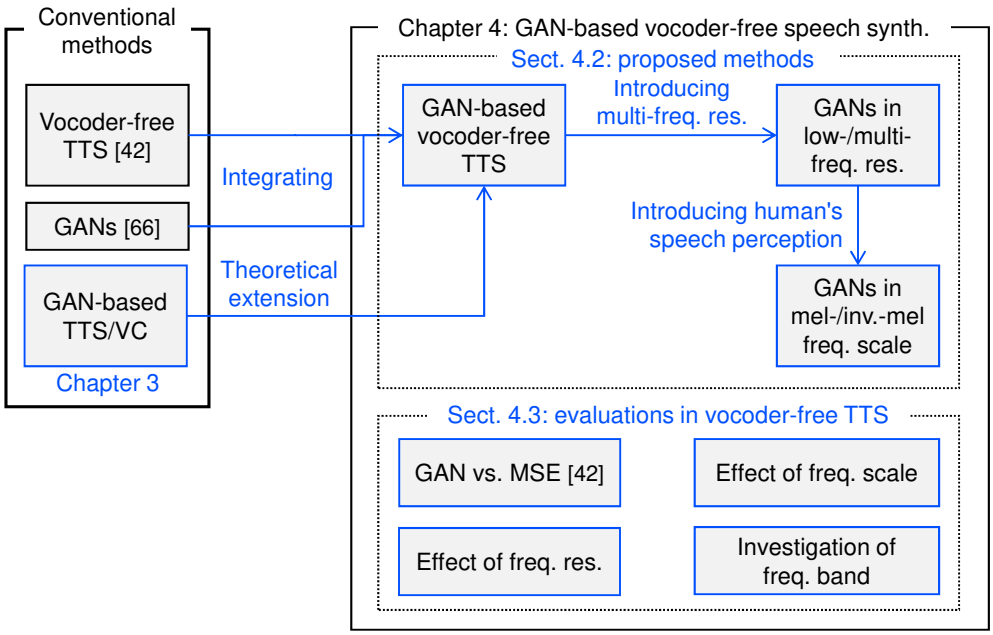

Fig. 4.3. Overview of Chapter 4

4.2 GAN-based Acoustic Model Training Using Low-/multi-frequency-resolution Amplitude Spectra

4.2.1 GAN-based Training Algorithm Using Low-frequency-resolution Amplitude Spectra

Frequency Resolution Lowering Using Average-pooling

Let $\boldsymbol{P} = [\boldsymbol{O}_{p,F}^\top \ \boldsymbol{I}_F^\top \ \boldsymbol{O}_{p,F}^\top]^\top$ be a $(F+2p)T$-by-FT zero-padding matrix, where p, $\boldsymbol{O}_{p,F}$, and $\boldsymbol{I}_F$ denote the size of zero padding, p-by-F zero matrix, and F-by-F identity matrix, respectively. A $(F+2p)$-dimensional zero-padded amplitude spectra vector at frame t, $[\boldsymbol{0}_p^\top, \boldsymbol{y}_t^\top, \boldsymbol{0}_p^\top]^\top$, is calculated as $\boldsymbol{P}\boldsymbol{y}_t$, where $\boldsymbol{0}_p$ denotes a p-dimensional zero vector. Let $\boldsymbol{W}$ be a $F^{(\mathrm{L})}$-by-$(F+2p)$ pooling matrix. The $F^{(\mathrm{L})}$ is the total number of frequency bins calculated as

$$F^{(\mathrm{L})} = \frac{F+2p-w}{s} + 1, \tag{4.1}$$

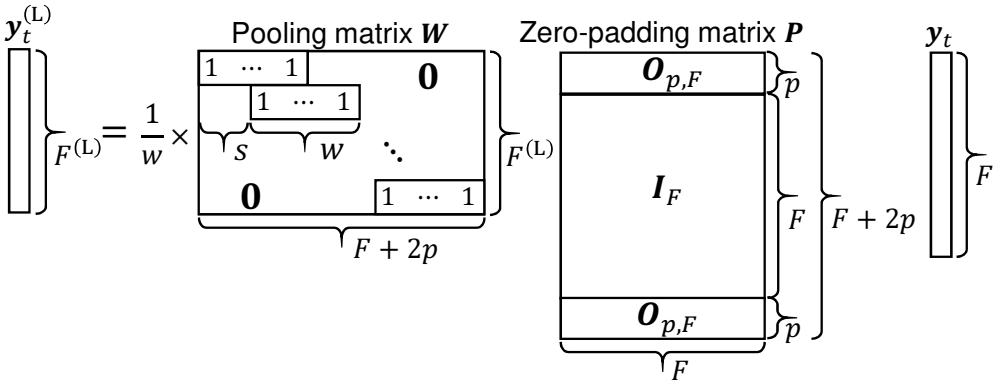

Fig. 4.4. Matrix computation in average-pooling function

where w and s denote the width and stride of the pooling operation, respectively. A low-frequency-resolution amplitude spectra vector at frame t, $\boldsymbol{y}_t^{(\mathrm{L})} = [y_t^{(\mathrm{L})}(1), \cdots, y_t^{(\mathrm{L})}(F^{(\mathrm{L})})]^\top$, is calculated as $\boldsymbol{W}\boldsymbol{P}\boldsymbol{y}_t$. An average-pooling function $\boldsymbol{\phi}(\cdot)$ transforms amplitude spectra in the original frequency resolution $\boldsymbol{y}$ into those in the low frequency resolution $\boldsymbol{y}^{(\mathrm{L})}$ as $\boldsymbol{\phi}(\boldsymbol{y}) = [(\boldsymbol{W}\boldsymbol{P}\boldsymbol{y}_1)^\top, \cdots, (\boldsymbol{W}\boldsymbol{P}\boldsymbol{y}_T)^\top]^\top$, using the matrices $\boldsymbol{P}$ and $\boldsymbol{W}$. Figure 4.4 shows the matrix computation in the average-pooling function. The above processes are similar to conversion from raw amplitude spectra into filter-bank parameters that represent spectral envelopes of speech.

GAN-based Training Using Low-frequency-resolution Amplitude Spectra

A low-frequency-resolution discriminator $D^{(\mathrm{L})}(\cdot)$ is trained to distinguish natural and generated amplitude spectra in the low (i.e., rough) frequency resolution. The discriminator loss for $D^{(\mathrm{L})}(\cdot)$ is defined as the same manner as Eq. (3.1), but $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$ in the equation are replaced with $\boldsymbol{y}^{(\mathrm{L})}$ and $\hat{\boldsymbol{y}}^{(\mathrm{L})}$, respectively.

A loss function for training an acoustic model is defined as follows:

$$L_{\mathrm{G}}^{(\mathrm{Low})}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = L_{\mathrm{MSE}}(\boldsymbol{y}, \hat{\boldsymbol{y}}) + \omega_{\mathrm{D}}^{(\mathrm{L})} \frac{\mathbb{E}_{\hat{\boldsymbol{y}}}[L_{\mathrm{MSE}}]}{\mathbb{E}_{\hat{\boldsymbol{y}}^{(\mathrm{L})}}[L_{\mathrm{ADV}}]} L_{\mathrm{ADV}}\left(\hat{\boldsymbol{y}}^{(\mathrm{L})}\right), \tag{4.2}$$

where $\hat{\boldsymbol{y}}^{(\mathrm{L})} = \boldsymbol{\phi}(\hat{\boldsymbol{y}})$ denotes generated spectra in the low frequency resolution. The $\omega_{\mathrm{D}}^{(\mathrm{L})}$ is a hyperparameter to control the effect of the second term. This loss function is formulated as the weighted sum of the MSE in the original frequency resolution and adversarial loss in the low frequency resolution. Since the distributions of amplitude spectra in the low frequency resolution become simpler than those in the original frequency resolution, this algorithm can overcome the difficulty in modeling complicated distribution of high-dimensional amplitude spectra. Furthermore, it can be expected to improve synthetic

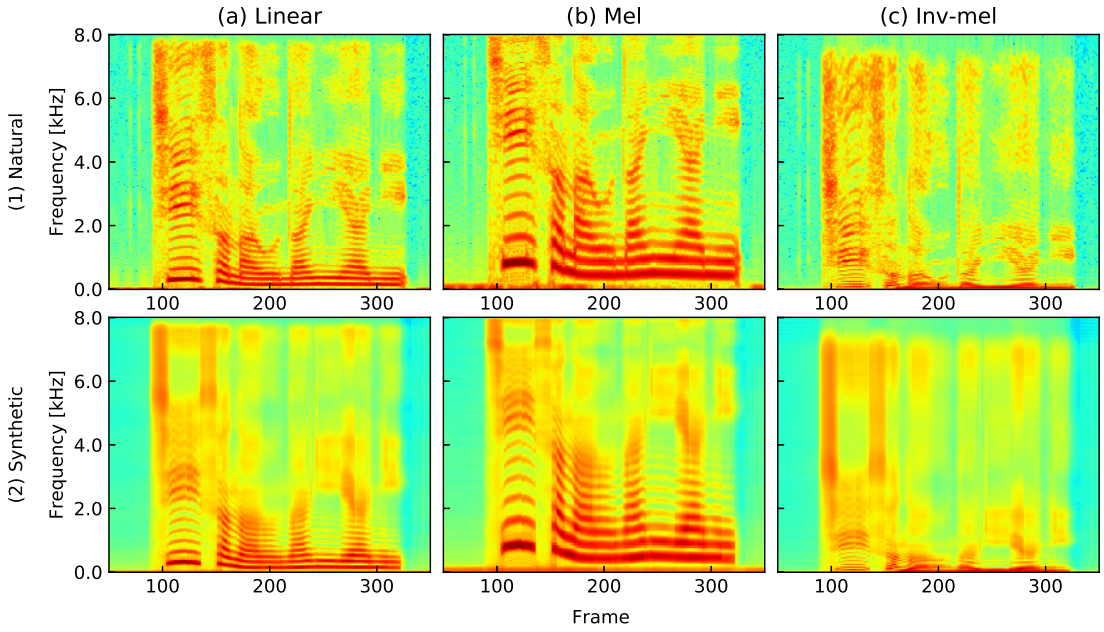

Fig. 4.5. Examples of amplitude spectra of natural and synthetic speech after frequency warping for (a) linear, (b) mel, and (c) inverse mel frequency scales. These spectra were extracted from one utterance of evaluation data. Synthetic amplitude spectra were generated from acoustic model trained to minimize MSE (Eq. (2.15)).

speech quality by reducing the difference between spectral envelopes (i.e., dominant features regarding speech quality) of natural and synthetic speech.

4.2.2 Frequency Scale Conversion Using Frequency Warping

Besides the above-described approximated filter bank extraction, the proposed algorithm can use different frequency scales that are related to human speech perception and anti-spoofing [29, 67]. For example, the mel frequency scale and its inverted version [68] can be used instead of an ordinary linear frequency scale. This can be done by applying a frequency warping function to amplitude spectra before feeding them into the low-frequency-resolution discriminator. Figure 4.5 shows examples of the amplitude spectra of natural and synthetic speech in various frequency scales after applying the frequency warping functions [102] shown in Fig. 4.6. In the mel frequency scale (Fig. 4.5(b)), there are fewer differences between natural and synthetic amplitude spectra than those in the inverse mel frequency scale. This observation suggests that the GAN-based distribution compensation may work well in the inverse mel frequency scale.

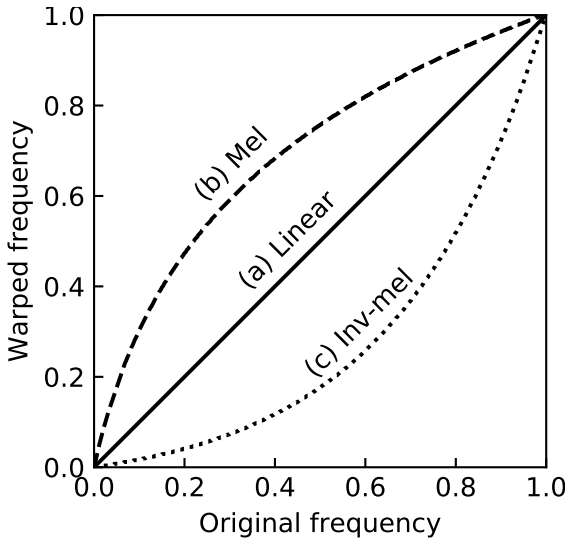

Fig. 4.6. Frequency warping functions for (a) linear, (b) mel, and (c) inverse mel frequency scales

4.2.3 GAN-based Training Algorithm Using Multi-frequency-resolution Amplitude Spectra

This section presents a GAN-based training algorithm using multi-frequency-resolution amplitude spectra. This algorithm introduces not only the low-frequency-resolution discriminator $D^{(\mathrm{L})}(\cdot)$ but also the original-frequency-resolution one $D(\cdot)$. A loss function for training an acoustic model is defined as follows:

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

When $\omega_{\mathrm{D}} = 0$, this loss function is the same as that in Eq. (4.2). This algorithm can be expected to compensate for not only the differences in *rough* structures (i.e., spectral envelopes) but also in *fine* structures of natural and generated amplitude spectra. Figure 4.7 illustrates the computation procedure of the loss function. Note that $D^{(\mathrm{L})}(\cdot)$ and $D(\cdot)$ are separately trained.

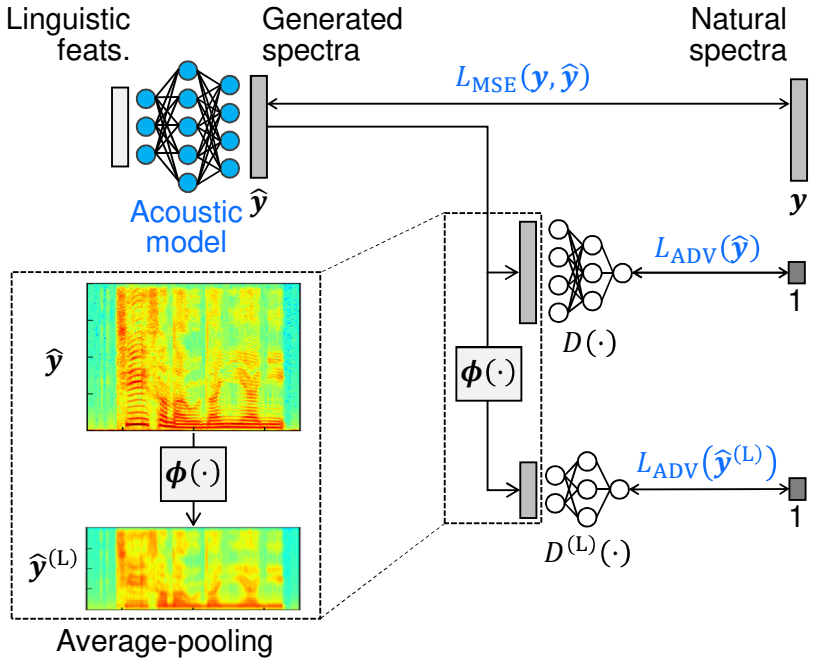

Fig. 4.7. Loss functions to update acoustic model in proposed GAN-based algorithm using multi-frequency-resolution amplitude spectra. Average-pooling function $\boldsymbol{\phi}(\cdot)$ lowers frequency resolution of amplitude spectra while keeping their rough structures unchanged.

4.2.4 Discussion

The average-pooling function in the proposed algorithms can be regarded as the filter bank parameter extraction. When the pooling width w is set to a larger value, fine structures of the amplitude spectra get smoother. Figure 4.8 shows examples of the low-frequency-resolution spectra of natural and synthetic speech with various settings of the pooling width. This figure illustrates that spectral peaks (i.e., formants) of the synthetic amplitude spectra tend to be weaker than those of the natural ones. This tendency may be one of the causes of the speech quality degradation.

Regarding related work, Kaneko et al. [103] proposed a GAN-based post-filter for STFT amplitude spectra. This post-filter-based approach requires additional computation in the synthesis stage, but the proposed approaches do not. Also, their approach ignores the overall spectral structures (i.e., spectral envelopes) and their correlation because it splits amplitude spectra into several sub-frequency bands and applies GANs to each band *independently*. On the other hand, the proposed approaches can effectively capture them by reducing the dimensionality of the spectra while preserving the whole spectral structure.

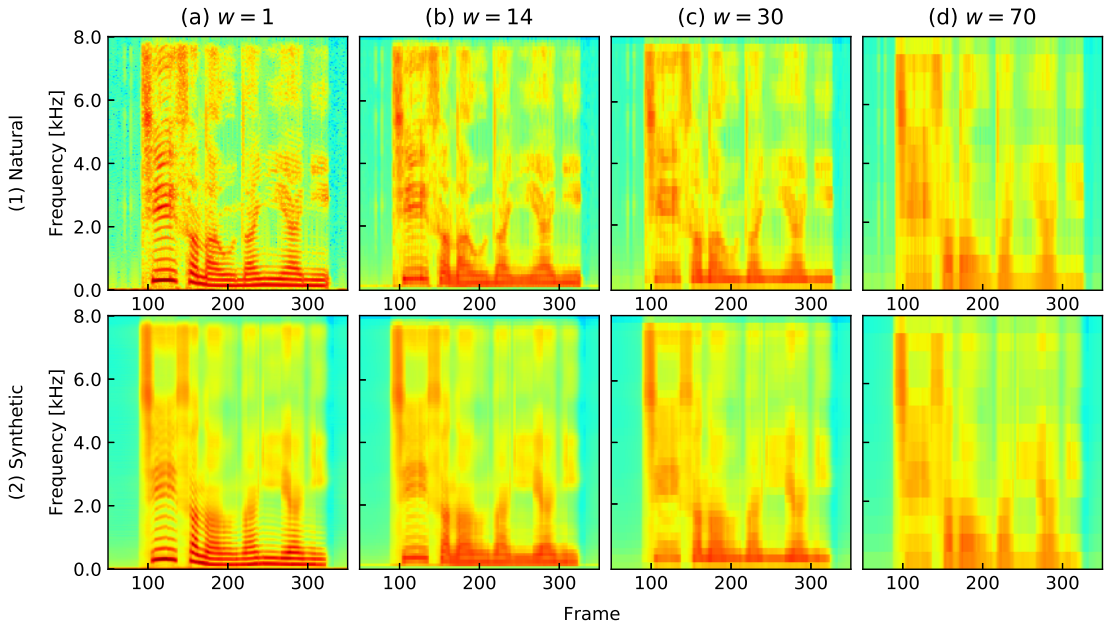

Fig. 4.8. Examples of low-frequency-resolution spectra of natural and synthetic speech. These spectra were extracted from one utterance of evaluation data. Zero-padding size p and pooling stride s were set to 6 and $w/2$, respectively. "(a) $w = 1$" corresponds to amplitude spectra in original frequency resolution. These synthetic amplitude spectra were generated from acoustic model trained to minimize Eq. (2.15).

This chapter extends the GAN-based proposed algorithm for TTS using *vocoder parameters* in Chapter 3 to TTS using *STFT amplitude spectra*. Although Juvela et al. [104] proposed a GAN-based method to synthesize a speech waveform from natural mel frequency cepstral coefficients (MFCCs), their method was not conditioned by linguistic information and cannot be directly applied to TTS. The proposed algorithms can be expected to achieve TTS that directly synthesizes a speech waveform from linguistic features. The idea using GANs in the low frequency resolution can also be applied to WaveGAN and SpecGAN [105], which synthesize an audio waveform by using unconditional GANs.

4.3 Experimental Evaluations

4.3.1 Experimental Conditions

A speech corpus of a Japanese female speaker was used. The speaker uttered 4,007 sentences (part of the JSUT corpus [106]). The numbers of sentences used for training and evaluation were 3,808 and 199, respectively. The sampling rate of the speech signals was 16 kHz. The frame length, shift length, and FFT length were 400, 80, and 1,024

samples, respectively. The Hamming window was used for STFT analysis. Accordingly, 513-dimensional STFT spectra were obtained. Ninety percent of the silence frames were removed from the training data to improve the training accuracy.

Architectures for the acoustic model and discriminators were Feed-Forward DNNs. The input of the acoustic model was a 444-dimensional vector including 439-dimensional linguistic features, 3-dimensional duration features, continuous log F_0, and U/V. The linguistic features included phonemes, mora position, accent type, frame position in a phoneme, etc. F_0 was extracted from speech data using the STRAIGHT vocoder [47]. Two DNNs for predicting duration and F_0 from linguistic features were trained in advance. The acoustic model included three 1,024-unit hidden layers with the ReLU [54] activation function and 513-unit output layer with the linear activation function. The original-frequency-resolution discriminator included three 512-unit hidden layers with the ReLU activation function and one unit output layer with the sigmoid activation function. The low-frequency-resolution discriminator was almost the same as the original-frequency-resolution one; i.e., the activation functions used in the hidden and output layers were ReLU and sigmoid, respectively, and the number of hidden layers was three. However, the numbers of input and hidden units of the discriminator varied in accordance with the parameters of the pooling function $\phi(\cdot)$. The pooling width w was set to 14, 30, or 70. Accordingly, $F^{(\mathrm{L})}$ was set to 74, 34, or 14. The number of the hidden units in $D^{(\mathrm{L})}(\cdot)$ was changed to 128, 64, or 32 as $F^{(\mathrm{L})}$ decreased. The zero padding size p and stride s were set to 6 and $w/2$, respectively.

In the DNN training, real-valued linguistic features and log-amplitude spectra were normalized to have zero-mean and unit-variance. The acoustic model was first initialized by minimizing the MSE between natural and generated amplitude spectra with 25 iterations. The original- and low-frequency-resolution discriminators were then initialized to distinguish amplitude spectra of natural speech and ones generated by the initialized acoustic model. The discriminator initialization was performed with 5 iterations. The proposed algorithms were finally performed with 25 iterations using the initialized acoustic model and discriminators. The expectation values for scaling the loss functions were estimated at each iteration. The optimization algorithm was AdaGrad [98]. The learning rate was set to 0.01.

4.3.2 Objective Evaluations

The root mean square error (RMSE) between natural and generated amplitude spectra and the spoofing rates of $D(\cdot)$ and $D^{(\mathrm{L})}(\cdot)$ were calculated as the objective evaluation criteria.

Table 4.1. Objective evaluation results with their standard deviations. "RMSE" denotes root mean square error between natural and generated amplitude spectra. Various hyperparameter settings $(\omega_{\mathrm{D}}, \omega_{\mathrm{D}}^{(\mathrm{L})})$ for proposed algorithms were used and compared. Pooling parameters were set to $w = 30$, $s = 15$, and $p = 6$. These evaluation values were calculated using all evaluation data and averaged

$(\omega_{\mathrm{D}}, \omega_{\mathrm{D}}^{(\mathrm{L})})$	(0.0, 0.0)	(0.0, 1.0)	(1.0, 0.0)	(1.0, 1.0)
RMSE	1.09±0.09	1.12±0.09	1.25±0.08	1.24±0.08
$D(\cdot)$ spoofing rate	0.0019 ± 0.0042	0.5507 ± 0.0696	0.9999 ± 0.0001	0.9999 ± 0.0005
$D^{(\mathrm{L})}(\cdot)$ spoofing rate	0.0273 ± 0.0177	0.9704 ± 0.0269	0.9965 ± 0.0055	0.9955 ± 0.0065

The discriminators $D(\cdot)$ and $D^{(\mathrm{L})}(\cdot)$ for the spoofing rate calculation were trained to distinguish amplitude spectra of natural speech from those of synthetic speech generated by the conventional algorithm [42]. The proposed algorithms were compared using the combination of the hyperparameters $(\omega_{\mathrm{D}}, \omega_{\mathrm{D}}^{(\mathrm{L})})$ setting each to 0.0 or 1.0.

Table 4.1 shows the evaluation results. From the results, the algorithm with the hyperparameter setting $(\omega_{\mathrm{D}} = 0.0, \omega_{\mathrm{D}}^{(\mathrm{L})} = 0.0)$, i.e., the same as the conventional algorithm, achieves the lowest RMSE. However, the algorithm's spoofing rates are the lowest among the four algorithms. This result suggests that the conventional algorithm does not train the acoustic model to fool the two discriminators. On the other hand, the proposed algorithm setting to $(\omega_{\mathrm{D}} = 0.0, \omega_{\mathrm{D}}^{(\mathrm{L})} = 1.0)$ generates amplitude spectra that fool $D^{(\mathrm{L})}(\cdot)$, although the RMSE becomes slightly worse than that of the conventional algorithm. Note that the similar tendency (i.e., increase of speech parameter generation error) was also reported in Section 3.4.2. Some of amplitude spectra generated by this algorithm also fool $D(\cdot)$, nevertheless deceiving the model is out of the consideration during the acoustic model training. These results indicate that the GAN-based proposed algorithm using low-frequency-resolution amplitude spectra reduces the differences between natural and generated amplitude spectra observed in their rough structures. Meanwhile, the proposed algorithms using original-frequency-resolution amplitude spectra, i.e., "(1.0, 0.0)" and "(1.0, 1.0)" in Table 4.1, have similar tendencies. Although these algorithms improve spoofing rates of both $D(\cdot)$ and $D^{(\mathrm{L})}(\cdot)$ much higher than the other two algorithms, they also considerably degrade the RMSE of the amplitude spectra. Such RMSE degradation may deteriorate synthetic speech quality significantly.

Table 4.2. Preference scores of synthetic speech quality (GANs using original-frequency-resolution amplitude spectra with various hyperparameter settings of ω_{D}). **Bold** values indicate that method is more preferred than other with p-value < 0.05

Method A	Score (A vs. B)	Method B
$\omega_{\mathrm{D}} = 0.5$	0.300 vs. **0.700**	Baseline
$\omega_{\mathrm{D}} = 1.0$	0.280 vs. **0.720**	Baseline
$\omega_{\mathrm{D}} = 0.5$	0.496 vs. 0.504	$\omega_{\mathrm{D}} = 1.0$

4.3.3 Subjective Evaluations

Preference AB tests in terms of synthetic speech quality were conducted. Twenty-five listeners participated in each of the following evaluations by using crowdsourced evaluation systems. Each listener evaluated 10 speech samples. The total number of listeners was 1,125. In the following evaluations, "Baseline" denotes the conventional training algorithm that minimizes the MSE loss shown in Eq. (2.15) only [42].

Evaluation of GANs Using Original-frequency-resolution Amplitude Spectra
The effect of the GAN-based algorithm using the original-frequency-resolution amplitude spectra (i.e., the same algorithm as described in Chapter 3) was investigated by fixing $\omega_{\mathrm{D}}^{(\mathrm{L})} = 0$ and by setting $\omega_{\mathrm{D}} = 0.5$ or 1.0. Three algorithms were compared: "Baseline" and proposed algorithm with the setting "$\omega_{\mathrm{D}} = 0.5$" or "$\omega_{\mathrm{D}} = 1.0$."

Table 4.2 shows the evaluation results. The GAN-based algorithm significantly degrades synthetic speech quality compared with "Baseline," regardless of the hyperparameter settings. This result demonstrates that just using the GAN-based training algorithm, which is effective in TTS using vocoder parameters (Chapter 3), does not improve synthetic speech quality in TTS using STFT amplitude spectra.

Evaluation of GANs Using Low-frequency-resolution Amplitude Spectra
Firstly, the effect of pooling width w was investigated by fixing $\omega_{\mathrm{D}} = 0$ and by setting $\omega_{\mathrm{D}}^{(\mathrm{L})} = 1$. Speech samples generated by "Baseline" and the proposed algorithm using low-frequency-resolution amplitude spectra were compared. The pooling width w was set to 14, 30, or 70. The number of hidden units in $D^{(\mathrm{L})}(\cdot)$ was changed in accordance with the pooling parameter settings. Table 4.3 shows the evaluation results. The proposed algorithm always achieves higher scores than "Baseline" regardless of the pooling width settings. This result demonstrates the algorithm's effectiveness in improving synthetic

Table 4.3. Preference scores of synthetic speech quality (GANs using low-frequency-resolution amplitude spectra with various settings of w). **Bold** values indicate that method is more preferred than other with p-value < 0.05. Here, number of hidden units in $D^{(\mathrm{L})}(\cdot)$ was changed in accordance with settings of w

(a) Results comparing "Baseline" with each of GAN-based algorithms

Method A	Score (A vs. B)	Method B
$w = 14$	**0.568** vs. 0.432	Baseline
$w = 30$	**0.572** vs. 0.428	Baseline
$w = 70$	0.528 vs. 0.472	Baseline

(b) Results comparing three GAN-based algorithms

Method A	Score (A vs. B)	Method B
$w = 14$	0.488 vs. 0.512	$w = 30$
$w = 30$	0.532 vs. 0.468	$w = 70$
$w = 70$	0.472 vs. 0.528	$w = 14$

Table 4.4. Preference scores of synthetic speech quality (GANs using low-frequency-resolution amplitude spectra with various settings of w). **Bold** values indicate that method is more preferred than other with p-value < 0.05. Here, number of hidden units in $D^{(\mathrm{L})}(\cdot)$ was fixed regardless of settings of w

(a) Results comparing "Baseline" with each of GAN-based algorithms

Method A	Score (A vs. B)	Method B
$w = 14$	**0.548** vs. 0.452	Baseline
$w = 30$	**0.600** vs. 0.400	Baseline
$w = 70$	**0.560** vs. 0.440	Baseline

(b) Results comparing three GAN-based algorithms

Method A	Score (A vs. B)	Method B
$w = 14$	0.472 vs. 0.528	$w = 30$
$w = 30$	0.528 vs. 0.472	$w = 70$
$w = 70$	0.476 vs. 0.524	$w = 14$

speech quality.

Secondly, a subjective evaluation of the proposed algorithm with the fixed number of hidden units in $D^{(\mathrm{L})}(\cdot)$ was conducted. The number of hidden units was set to 128 regardless of the pooling parameter settings. Table 4.4 shows the evaluation results. The results shown in this table have tendencies similar to Table 4.3, demonstrating that the

Table 4.5. Preference scores of synthetic speech quality (GANs using low-frequency-resolution amplitude spectra with various hyperparameter settings of $\omega_{\mathrm{D}}^{(\mathrm{L})}$ and fixed pooling width $w = 30$). **Bold** values indicate that method is more preferred than other with p-value < 0.05

Method A	Score (A vs. B)	Method B
$\omega_{\mathrm{D}}^{(\mathrm{L})} = 0.5$	**0.544** vs. 0.456	Baseline
$\omega_{\mathrm{D}}^{(\mathrm{L})} = 1.0$	**0.588** vs. 0.412	Baseline
$\omega_{\mathrm{D}}^{(\mathrm{L})} = 0.5$	0.504 vs. 0.496	$\omega_{\mathrm{D}}^{(\mathrm{L})} = 1.0$

Table 4.6. Preference scores of synthetic speech quality (comparison between GANs using low-frequency-resolution and smoothed original-frequency-resolution amplitude spectra). **Bold** value indicates that method is more preferred than other with p-value < 0.05

Method A	Score (A vs. B)	Method B
Pooling	**0.580** vs. 0.420	Smoothing

algorithm works robustly against the discriminator's size. The pooling width w was set to 30 in the following evaluations because the results in Tables 4.3(b) and 4.4(b) indicated that "$w = 30$" was the best among the three settings. The number of hidden units in $D^{(\mathrm{L})}(\cdot)$ was set to 64 for reducing the number of model parameters.

Finally, the hyperparameter's effect in the proposed algorithm was investigated by fixing $\omega_{\mathrm{D}} = 0$ and by setting $\omega_{\mathrm{D}}^{(\mathrm{L})} = 0.5$ or 1.0. "Baseline" and the proposed algorithm with the setting "$\omega_{\mathrm{D}}^{(\mathrm{L})} = 0.5$" or "$\omega_{\mathrm{D}}^{(\mathrm{L})} = 1.0$" were compared. Table 4.5 shows the evaluation results. These results demonstrate that the GAN-based proposed algorithm using low-frequency-resolution amplitude spectra improves synthetic speech quality regardless of its hyperparameter settings.

Comparison between Pooling and Smoothing of Amplitude Spectra

The dimensionality reduction by the average-pooling function is one of the crucial factors in the proposed algorithm. Here, the average-pooling function ("Pooling") was compared with a simple moving average filter ("Smoothing") to investigate the effectiveness of the dimensionality reduction. The filter size of "Smoothing" was set to 25. Amplitude spectra after applying "Smoothing" can be regarded as the spectral envelope parameters without the dimensionality reduction. Table 4.6 shows the evaluation result. "Pooling" significantly outperforms "Smoothing" from this table. This result suggests that the dimensionality reduction is one of the essentials for the synthetic speech quality improvement.

Table 4.7. Preference scores of speech quality (GANs using original-, low-, and multi-frequency-resolution amplitude spectra). **Bold** values indicate that method is more preferred than other with p-value < 0.05

Method A	Score (A vs. B)	Method B
Low	**0.808** vs. 0.192	Multi
Multi	0.492 vs. 0.508	Original
Original	0.192 vs. **0.808**	Low

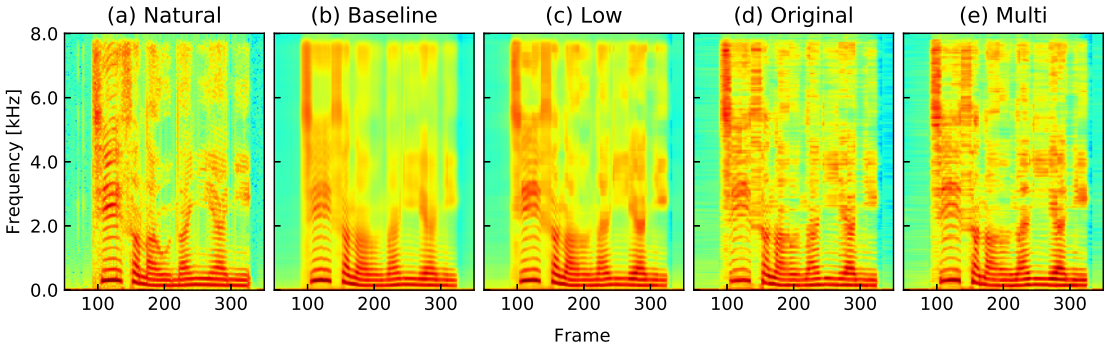

Fig. 4.9. Examples of amplitude spectra of (a) natural speech and synthetic speech generated by algorithms named as (b) "Baseline," (c) "Low," (d) "Original," and (e) "Multi." These spectra were extracted from one utterance of evaluation data.

Evaluation of GANs Using Multi-frequency-resolution Amplitude Spectra

The effects of the proposed algorithm using multi-frequency-resolution amplitude spectra were investigated. Speech samples were generated using the following three 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 4.7 shows the experimental results. "Low" in this table significantly outperforms the others. Amplitude spectra of synthetic speech used for this evaluation were plotted in Fig. 4.9 to investigate the reason. The difference in spectral peaks between natural and synthetic speech is reduced by the proposed algorithms (Figs. 4.9(c), (d), and (e)). However, some temporal discontinuities are observed in the amplitude spectra generated by "Original" and "Multi" (Figs. 4.9(d) and (e)). These discontinuities may be one of the causes of speech quality degradation.

Table 4.8. Preference scores of synthetic speech quality (GANs using low-frequency-resolution amplitude spectra with gated-CNN-based acoustic model). **Bold** values indicate that method is more preferred than other with p-value < 0.05

(a) Results comparing "Baseline" with each of GAN-based algorithms

Method A	Score (A vs. B)	Method B
Original	0.180 vs. **0.820**	Baseline
Low	**0.608** vs. 0.392	Baseline
Multi	0.252 vs. **0.748**	Baseline

(b) Results comparing three GAN-based algorithms

Method A	Score (A vs. B)	Method B
Low	**0.808** vs. 0.192	Multi
Multi	0.496 vs. 0.504	Original
Original	0.172 vs. **0.828**	Low

Evaluation of DNN Architectures for Acoustic model

Richer architectures than Feed-Forward DNNs were used for acoustic modeling to deal with the temporal discontinuities of generated amplitude spectra. Gated convolutional neural networks (CNNs) [107] were used instead of LSTM [52, 83] because they can be applied to sequential modeling in speech processing [108, 109] and can be trained faster than LSTM. Here, a 1D convolutional (Conv1D) layer along the time axis with the gated linear unit (GLU) activation function [107] was prepared after the output layer of the acoustic model. A residual connection [110] was introduced between the output and Conv1D layers for better modeling. The width and zero-padding size of the convolution were set to 15 and 7, respectively.

Speech samples were generated by the four algorithms ("Baseline," "Original," "Low," and "Multi") using the gated CNNs as the acoustic model. Table 4.8 shows the evaluation results. Similar tendencies to the previous evaluation results are observed; i.e., 1) "Low" significantly outperforms the others, and 2) the GAN-based algorithms using original-frequency-resolution amplitude spectra significantly degrade synthetic speech quality. These results indicate that the sequential modeling is insufficient to deal with the temporal discontinuities in the generated amplitude spectra.

The effectiveness of the CNN-based acoustic model in the proposed algorithm using low-frequency-resolution amplitude spectra was investigated. Two acoustic models, 1) Feed-Forward DNNs ("FFNN") and 2) gated CNNs ("CNN"), were compared. Table

Table 4.9. Preference scores of synthetic speech quality (comparison between Feed-Forward DNNs ("FFNN") and gated CNNs ("CNN") for acoustic modeling). **Bold** values indicate that method is more preferred than other with p-value < 0.05

Method A	Score (A vs. B)	Method B
CNN	**0.644** vs. 0.356	FFNN

Table 4.10. DNN architectures for discriminator. Numbers of input units of "FFNN-O," "FFNN-L," "CNN-1D," and "CNN-2D," were 513, 34, 513, and 513, respectively. "ReLU" or "GLU" denotes hidden activation function. Width, stride, and zero-padding size of Conv1D layer in "CNN-1D" were set to 30, 15, and 6, respectively. Width parameters of Conv2D layers in "CNN-2D" were set to 9, 7, 5, and 3. Accordingly, stride and zero-padding parameters were set to 4, 3, 2, and 1

	Hidden layers	Output
FFNN-O	Linear 512 units × 3 (ReLU)	Frame-wise
FFNN-L	Linear 64 units × 3 (ReLU)	Frame-wise
CNN-1D	Conv1D 4 channels (GLU) + Linear 64 units × 3 (ReLU)	Frame-wise
CNN-2D	Conv2D 32–16–8–4 channels (ReLU) + Fully-connected	Segment-wise

4.9 shows the evaluation results. The results demonstrate that the CNN-based acoustic model is effective in improving synthetic speech quality better than the Feed-Forward-DNN-based one.

Evaluation of DNN Architectures for Discriminator

The effects of DNN architectures for the discriminator in the proposed algorithms were investigated. The gated-CNN-based acoustic model was used, and the discriminators using following four DNN architectures were compared:

FFNN-O: the same as used in "Original"

FFNN-L: the same as used in "Low"

CNN-1D: replacing average-pooling with a Conv1D layer along the frequency axis with the GLU activation function

CNN-2D: using 2D convolutional (Conv2D) layers for capturing the time-frequency structures of the spectra

The Conv1D layer in "CNN-1D" can be regarded as a trainable multi-channel average-pooling function to obtain low-frequency-resolution amplitude spectra. Table 4.10 shows the details of the four architectures.

Table 4.11. Preference scores of synthetic speech quality (GANs using gated CNNs as acoustic models and various DNN architectures for discriminative models). **Bold** values indicate that method is more preferred than other with p-value < 0.05

(a) Results comparing GANs using original-frequency-resolution amplitude spectra

Method A	Score (A vs. B)	Method B
FFNN-O	0.376 vs. **0.624**	CNN-1D
FFNN-O	0.120 vs. **0.880**	CNN-2D
CNN-1D	0.348 vs. **0.652**	CNN-2D

(b) Results comparing GAN-based algorithms using original- and low-frequency-resolution amplitude spectra

Method A	Score (A vs. B)	Method B
FFNN-L	**0.820** vs. 0.180	CNN-1D
FFNN-L	**0.828** vs. 0.172	CNN-2D

Speech samples were generated using the gated-CNN-based acoustic model trained with the four different discriminators. Table 4.11 shows the evaluation results. The results shown in Table 4.11(a) demonstrate that 1) both "CNN-1D" and "CNN-2D" outperform "FFNN-O," and 2) "CNN-2D" is superior to "CNN-1D," regarding synthetic speech quality. However, the results in Table 4.11(b) show that the CNN-based discriminators are inferior to "FFNN-L." One of the reasons may be the difficulty in training GANs with more complicated DNN architectures for the proposed algorithm.

Evaluation of Frequency Scale of Low-frequency-resolution Amplitude Spectra
The effect of the frequency scale in the GAN-based algorithm using low-frequency-resolution amplitude spectra was investigated. Three types of frequency scales, linear, mel, and inverse mel, were compared. Table 4.12 shows the evaluation results. "Mel" significantly degrades synthetic speech quality, while "Inv-mel" successfully improves it and even outperforms "Linear." Amplitude spectra of synthetic speech used for this evaluation were plotted in Fig. 4.10 to investigate the reason. The differences among the three generated amplitude spectra are observed in a higher frequency band in particular. The generated spectra of "Low w/ inv-mel" become closer to the natural spectra compared to the others. This effect may by one of the causes of the quality improvement.

Table 4.12. Preference scores of synthetic speech quality (GANs using low-frequency-resolution amplitude spectra with various frequency scale). **Bold** values indicate that method is more preferred than other with p-value < 0.05

(a) Results comparing "Baseline" with GANs using low-frequency-resolution amplitude spectra with mel or inverse mel frequency scale

Method A	Score (A vs. B)	Method B
Mel	0.392 vs. **0.608**	Baseline
Inv-mel	**0.636** vs. 0.364	Baseline

(b) Results comparing three GAN-based algorithms with different frequency scale

Method A	Score (A vs. B)	Method B
Linear	**0.752** vs. 0.248	Mel
Mel	0.272 vs. **0.728**	Inv-mel
Inv-mel	**0.576** vs. 0.424	Linear

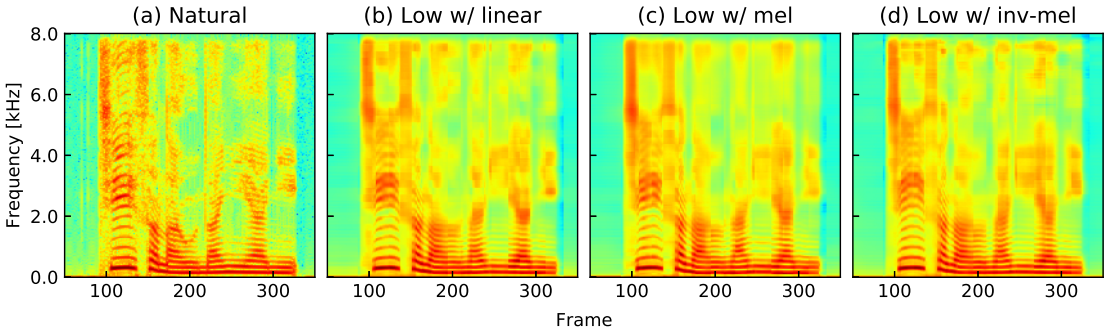

Fig. 4.10. Examples of amplitude spectra of (a) natural speech and synthetic speech generated by proposed algorithms using GANs in low frequency resolution with (b) linear scale, (c) mel scale, and (d) inverse mel scales. These spectra were extracted from one utterance of evaluation data.

Evaluation of Frequency Band Used for Average-pooling

The average-pooling function in the proposed algorithm can be regarded as the reduction of undesired components in amplitude spectra fed into the discriminator. Here, the spectra were first split into two sub-bands: 0–2 kHz (including at least the first and second formants) and 2–8 kHz. The average-pooling was then applied to each sub-band individually. The results of the comparison among the combinations of these two factors: frequency band (low or high) and average-pooling (with or without), should be meaningful for clarifying what component is effective in improving synthetic speech quality. Figure 4.11 illustrates a conceptual diagram of the split-and-pooling procedures.

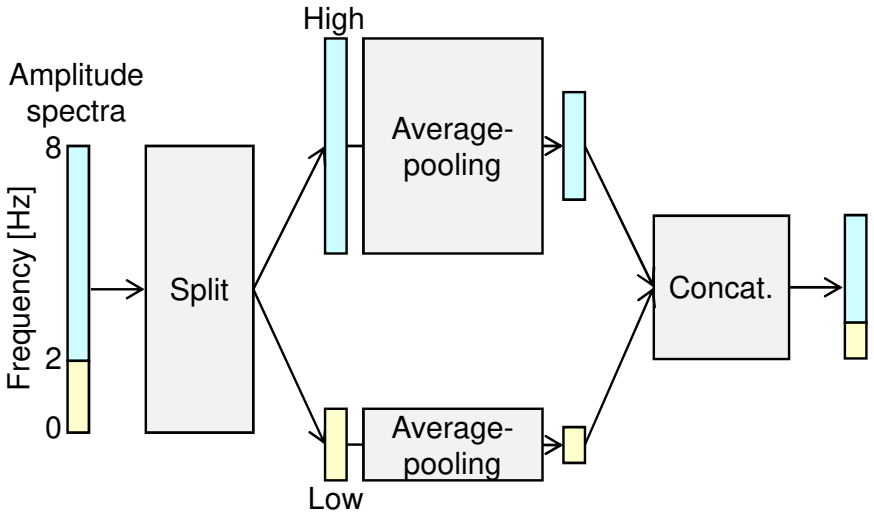

Fig. 4.11. Conceptual diagram of split-and-pooling procedures. This figure corresponds to proposed algorithm with average-pooling applied to both low and high frequency bands, i.e., "(w/, w/)" in Table 4.13.

Table 4.13. Preference scores of synthetic speech quality (comparison among the combination of frequency band (low or high) and average-pooling (w/ or w/o)). **Bold** values indicate that method is more preferred than other with p-value < 0.05

Method A (low, high)	Score (A vs. B)	Method A (low, high)
(w/o, w/o)	0.212 vs. **0.788**	(w/, w/)
(w/o, w/o)	0.160 vs. **0.840**	(w/, w/o)
(w/o, w/o)	**0.668** vs. 0.332	(w/o, w/)
(w/o, w/)	0.160 vs. **0.840**	(w/, w/o)
(w/, w/o)	0.468 vs. 0.532	(w/, w/)
(w/, w/)	**0.844** vs. 0.156	(w/o, w/)

Table 4.13 shows the evaluation results. Here, Feed-Forward DNNs were used for both acoustic model and discriminator. Two noteworthy points are observed in Table 4.13: 1) using average-pooling in the *low* frequency band always achieves higher scores, and 2) there is no significant difference between average-pooling in *both low and high* frequency bands and that in an *only low* frequency band. These results suggest that applying the GAN-based distribution compensation to the low frequency band tends to degrade the synthetic speech quality. This tendency corresponds to the results shown in Table 4.12(a) that the use of the mel frequency scale significantly deteriorates synthetic speech quality.

4.4 Summary

This chapter proposed two training algorithms to incorporate GANs into DNN-based vocoder-free speech synthesis using STFT spectra. The proposed algorithm using low-frequency-amplitude spectra trains an acoustic model to minimize the MSE between natural and generated amplitude spectra in original frequency resolution and the difference of their distributions in low frequency resolution. This algorithm can be extended to one using multi-frequency-resolution amplitude that also minimizes the distribution differences in the original frequency resolution. Experimental results showed that GANs using original-/multi-frequency-resolution amplitude spectra degraded synthetic speech quality, but using low-frequency-resolution amplitude spectra successfully improved it better than the conventional algorithm. Moreover, GANs using low-frequency-resolution amplitude spectra with the inverse mel frequency scale further improved speech quality.

Chapter 5

VAE-based Multi-speaker Acoustic Modeling Using Speech Recognition Process

5.1 Introduction

This chapter proposes a VAE-based multi-speaker speech synthesis method that can synthesize arbitrary speakers' high-quality voices and can transform the voice characteristics of the synthetic speech using a single model. Figure 5.1 shows a conceptual diagram of the proposed method. The proposed method trains a VAE-based acoustic model with the aid of a DNN-based speech recognition model that predicts phonetic content of input speech. As a result, the phonetic content of synthetic speech can be clarified and high quality speech can be synthesized. The proposed method also utilizes continuous speaker representations derived from a DNN-based speaker classification model to overcome the limitation in the use of conventional speaker codes (i.e., discrete speaker representations). Such continuous speaker representations enable establishing VAE-based non-parallel and many-to-many VC that can convert arbitrary speakers' voice characteristics into other arbitrary speakers' without requiring parallel speech corpora for the training. Figure 5.2 illustrates the relation between conventional and proposed methods.

This chapter is organized as follows (see also Fig. 5.3). Section 5.2 describes the proposed VAE-based multi-speaker speech synthesis method. Non-parallel and many-to-many VC based on this method is also presented. Section 5.3 presents experimental evaluations of this method in VC. Section 5.4 summarizes this chapter.

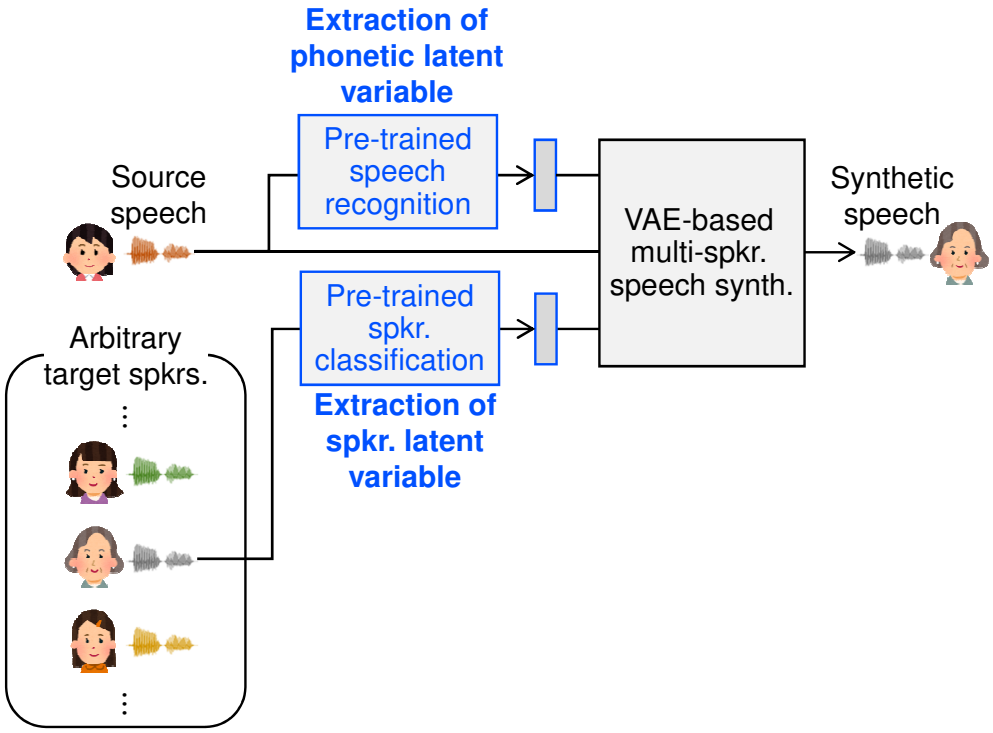

Fig. 5.1. Conceptual diagram of proposed method in Chapter 5

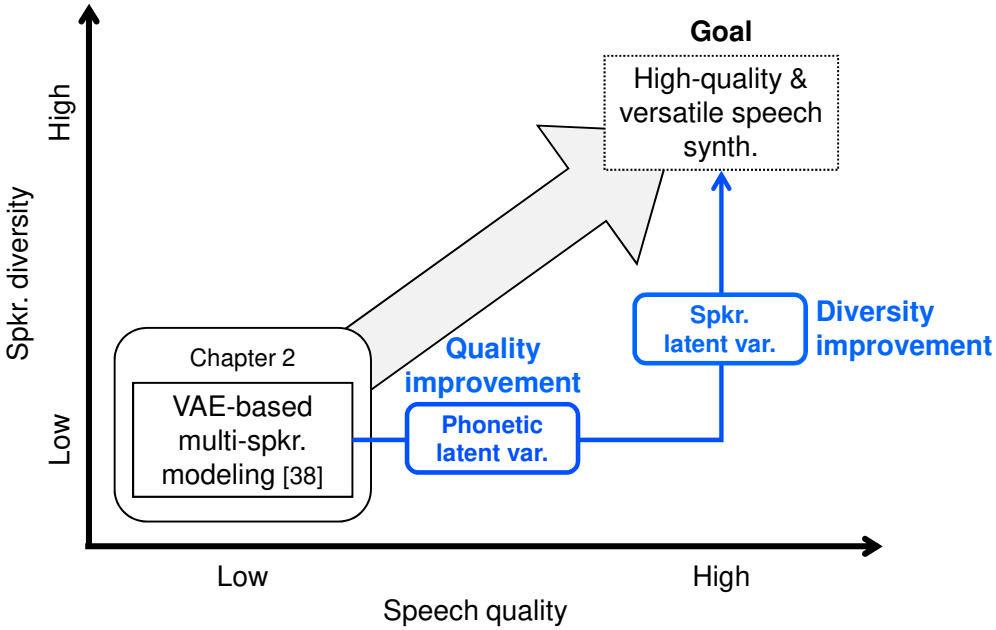

Fig. 5.2. Relation between conventional and proposed methods in Chapter 5

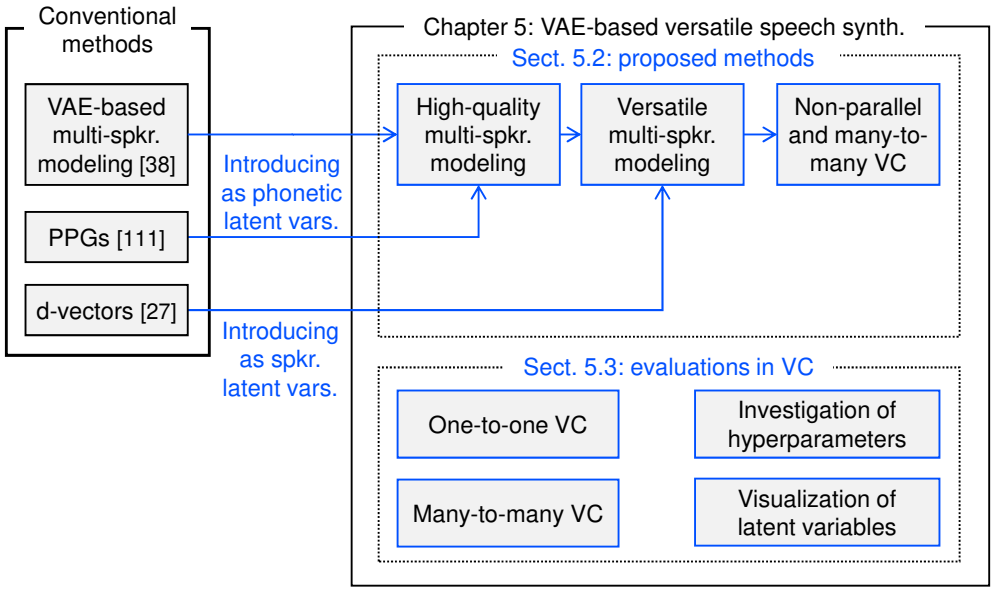

Fig. 5.3. Overview of Chapter 5

5.2 VAE-based Multi-speaker Acoustic Model Using Phonetic and Speaker Latent Variables

5.2.1 Phonetic Latent Variable to Improve Synthetic Speech Quality

VAE Training Objective Using Phonetic Latent Variable

A phonetic latent variable, derived from pre-trained DNNs for speech recognition, is introduced for alleviating quality degradation of synthetic speech due to over-regularization of VAE latent variables. Let $\boldsymbol{z}_{\mathrm{p}}$ be a phonetic latent variable extracted from speech parameters $\boldsymbol{y}$. The objective function in Eq. (2.14) is rewritten as:

$$\mathcal{L}\left(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{y}, \boldsymbol{c}, \boldsymbol{z}_{\mathrm{p}}\right) = -D_{\mathrm{KL}}\left(q_{\boldsymbol{\phi}}\left(\boldsymbol{z}|\boldsymbol{y}, \boldsymbol{z}_{\mathrm{p}}\right) || \, p_{\boldsymbol{\theta}}\left(\boldsymbol{z}\right)\right) + \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{y}, \boldsymbol{z}_{\mathrm{p}})}\left[\log p_{\boldsymbol{\theta}}\left(\boldsymbol{y}|\boldsymbol{z}, \boldsymbol{z}_{\mathrm{p}}, \boldsymbol{c}\right)\right]. \quad (5.1)$$

Here, $\boldsymbol{z}_{\mathrm{p}}$ is fed into both the encoder and decoder networks for keeping the phonetic content of input speech preserved. Figure 5.4(b) shows the directed graphical model of the proposed VAE-based acoustic model using phonetic latent variables and speaker codes.

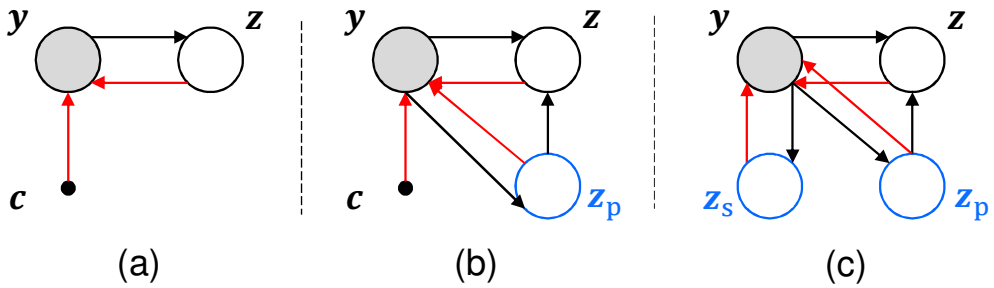

Fig. 5.4. Three directed graphical models of VAE-based multi-speaker acoustic models. From left, VAEs are conditioned by (a) one-hot speaker code $\boldsymbol{c}$, (b) $\boldsymbol{c}$ and phonetic latent variable $\boldsymbol{z}_{\mathrm{p}}$, and (c) speaker latent variable $\boldsymbol{z}_{\mathrm{s}}$ and $\boldsymbol{z}_{\mathrm{p}}$, respectively. Black and red arrows denote inference of latent variables $\boldsymbol{z}_{*}$ and generation of speech parameter $\boldsymbol{y}$, respectively.

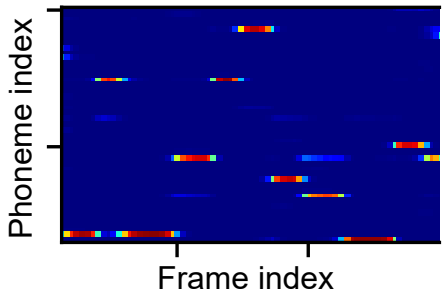

Fig. 5.5. Example of PPGs. Horizontal and vertical axes represent temporal axis and phoneme index, respectively. Brighter values denote high posterior probabilities.

Phonetic Posteriorgram (PPG) as Phonetic Latent Variable

A PPG [111], predicted by a DNN-based speech recognition model, is used as the phonetic latent variable. It represents a posterior probability sequence of phoneme labels given input speech parameters. Figure 5.5 shows an example of PPGs. The speech recognition model takes speech parameters as input and predicts a phoneme label that represents phonetic content of input speech. A loss function for the DNN training is defined as the softmax cross-entropy (SCE) of speech recognition:

$$L_{\mathrm{SCE}}\left(\boldsymbol{p}, \hat{\boldsymbol{p}}\right) = -\sum_{n=1}^{N_{\mathrm{p}}} p\left(n\right) \log \hat{p}\left(n\right), \tag{5.2}$$

where $\boldsymbol{p} = [p(1), \cdots, p(n), \cdots, p(N_{\mathrm{p}})]^{\top}$ and $\hat{\boldsymbol{p}} = [\hat{p}(1), \cdots, \hat{p}(n), \cdots, \hat{p}(N_{\mathrm{p}})]^{\top}$ are a phoneme label and output vector of the DNNs (i.e., PPG), respectively. The N_{p} denotes the number of phoneme set. Figure 5.6 shows the training procedure for speech

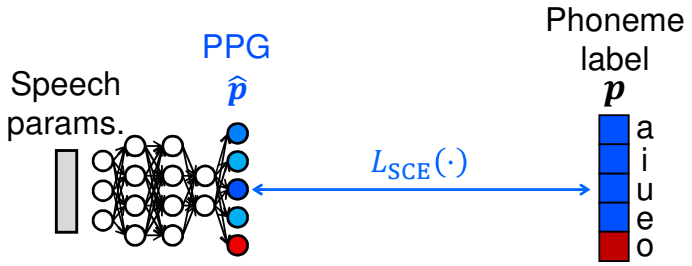

Fig. 5.6. Training procedure for DNN-based speech recognition model. PPG $\hat{\boldsymbol{p}}$ is output of DNNs.

recognition DNNs. The use of PPGs can be expected to improve synthetic speech quality since its effectiveness is well-known in training DNNs for many-to-one VC (Appendix B).

5.2.2 Speaker Latent Variable to Increase Speaker Diversity

VAE Training Objective Using Phonetic and Speaker Latent Variables

A speaker latent variable, derived from pre-trained DNNs for speaker classification, is used as a speaker representation to increase the speaker diversity in synthetic speech. It represents input speaker's individuality as a continuous vector even if the speaker is unseen (i.e., not included in training data). Let $\boldsymbol{z}_{\mathrm{s}}$ be a speaker latent variable extracted from speech parameters $\boldsymbol{y}$. The objective function Eq. (5.1) is rewritten as:

$$\mathcal{L}\left(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{y}, \boldsymbol{z}_{\mathrm{s}}, \boldsymbol{z}_{\mathrm{p}}\right) = -D_{\mathrm{KL}}\left(q_{\boldsymbol{\phi}}\left(\boldsymbol{z}|\boldsymbol{y}, \boldsymbol{z}_{\mathrm{p}}\right) || \, p_{\boldsymbol{\theta}}\left(\boldsymbol{z}\right)\right) + \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{y}, \boldsymbol{z}_{\mathrm{p}})}\left[\log p_{\boldsymbol{\theta}}\left(\boldsymbol{y}|\boldsymbol{z}, \boldsymbol{z}_{\mathrm{p}}, \boldsymbol{z}_{\mathrm{s}}\right)\right]. \tag{5.3}$$

Here, the conventional *discrete* speaker code $\boldsymbol{c}$ is replaced with the *continuous* speaker latent variable $\boldsymbol{z}_{\mathrm{s}}$. The use of continuous representations enables reproducing an unseen speaker's voice characteristics from his/her few speech utterances. Figures 5.4(c) and 5.7 show the directed graphical model and overview of the proposed VAE-based acoustic model using phonetic and speaker latent variables, respectively.

d-vector as Speaker Latent Variable

A d-vector [27], a bottleneck feature of a DNN-based speaker classification model, is used as the speaker latent variable. The speaker classification model takes speech parameters as input and predicts a one-hot speaker code $\boldsymbol{c} = [c(1), \cdots, c(n), \cdots, c(N_{\mathrm{s}})]^{\top}$ that represents the identity of one of the seen (i.e., pre-stored) N_{s} speakers. A loss function for the DNN

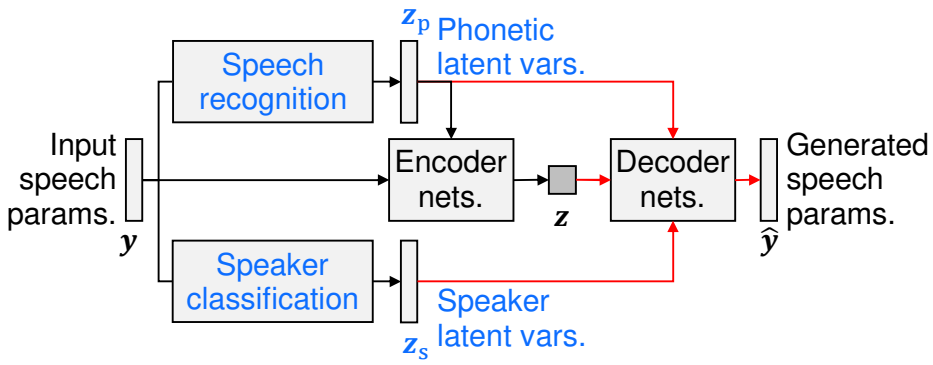

Fig. 5.7. Proposed VAE-based multi-speaker acoustic model integrating speech recognition and speaker classification models. This model corresponds to directed graphical model shown in Fig. 5.4(c).

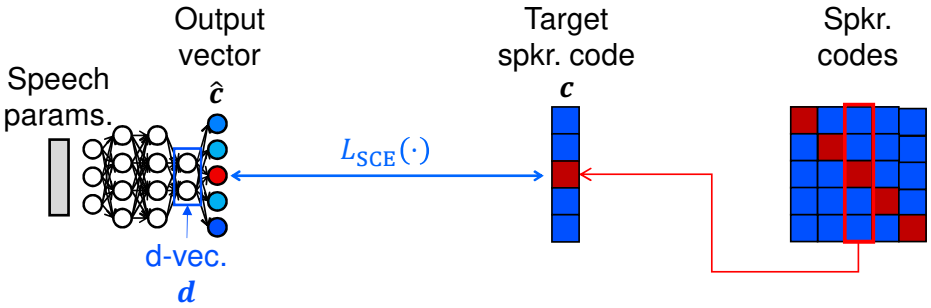

Fig. 5.8. Training procedure for DNN-based speaker classification model. d-vector $\boldsymbol{d}$ is extracted as output of squeeze layer immediately before output layer.

training is defined as the SCE of speaker classification:

$$L_{\mathrm{SCE}}\left(\boldsymbol{c}, \hat{\boldsymbol{c}}\right) = -\sum_{n=1}^{N_{\mathrm{s}}} c\left(n\right) \log \hat{c}\left(n\right), \tag{5.4}$$

where $\hat{\boldsymbol{c}} = [\hat{c}(1), \cdots, \hat{c}(n), \cdots, \hat{c}(N_{\mathrm{s}})]^{\top}$ is an output vector of the DNNs. Figure 5.8 shows the training procedure for speaker classification DNNs.

After the training, an N_{d}-dimensional d-vector $\boldsymbol{d} = [d(1), \cdots, d(N_{\mathrm{d}})]^{\top}$ is extracted from a bottleneck layer of the DNNs. One layer before the output is often used as the bottleneck layer. The d-vector dimensionality N_{d} is typically set to a smaller value than N_{s} for using the lower-dimensional speaker representation. One speaker's individuality can be defined by averaging all d-vectors extracted from his/her speech parameter sequences in *voiced* regions. This usage slightly differs from the traditional use of d-vectors for speaker verification that defines the speaker individuality as a d-vector averaged over those in *all* (i.e., voiced and unvoiced) regions. The reason why is that speech parameters in unvoiced

regions are less affected by speaker individuality than those in voiced regions [112, 57].

5.2.3 Non-parallel and Many-to-many VC Using Proposed VAE-based Acoustic Model

A non-parallel and many-to-many VC method can be established using the VAE-based multi-speaker acoustic model described in Section 5.2.2. The VAEs in this method are firstly trained using multiple speakers' speech utterances with the pre-trained DNNs for speech recognition and speaker classification. The target speaker's d-vector is secondly extracted from his/her speech parameters using the speaker classification DNNs. PPGs and VAE latent variables are thirdly predicted from source speaker's speech parameters using the speaker recognition DNNs and encoder networks of VAEs, respectively. The converted speech parameters are finally generated from the PPGs, VAE latent variables, and target speaker's d-vectors using decoder networks of VAEs.

5.2.4 Discussion

In the proposed method, DNNs for speech recognition and speaker classification must be pretrained using large speech corpora including many speakers. Although making transcriptions to train the speech recognition DNNs requires considerable cost, semi-supervised learning of conditional VAEs [58] can be used for reducing the cost. Also, techniques for end-to-end speech processing [113, 114] and dual learning of speech synthesis/recognition models [115, 116] can be applied to the proposed method. Furthermore, one can extend the proposed method to multilingual speech synthesis by introducing multilingual adaptation of DNN-based speech recognition [117].

The number of seen speakers and d-vector dimensionality are hyperparameters of the proposed method. The former should be large enough to learn better phonetic and speaker latent variables. The latter also affects the synthetic speech quality since it can be regarded as the number of basis in the speaker space. Section 5.3.4 empirically investigates the effect of the two hyperparameters.

5.3 Experimental Evaluation

5.3.1 Experimental Conditions

The effectiveness of the proposed method was evaluated in DNN-based VC. Two speech corpora were used: a parallel speech corpus including few speakers and non-parallel speech corpus including many speakers. The first included six Japanese speakers (three males and three females) who uttered 425 fully-parallel sentences. The second included 260 Japanese speakers (130 males and 130 females) who uttered about 100 non-parallel sentences. The sampling rate was 22.05 kHz.

The proposed method was evaluated in one-to-one VC to investigate the effectiveness of the phonetic latent variables (i.e., PPGs) in improving synthetic speech quality. DNN-based acoustic models were trained for each pair of source and target speakers taken from the first corpus. Since the three male and three female speakers were used, there were totally 12 inter-gender VC settings: six male-to-male (m2m) and six female-to-female (f2f), and 18 inter-gender ones: nine male-to-female (m2f) and nine female-to-male (f2m). The 1st-through-400th utterances were divided into two subsets for making a non-parallel VC setting: the half for source speakers and the remainders for target speakers. The parallel 401st-through-425th utterances were used for the evaluation.

The proposed method was also evaluated in many-to-many VC. Here, DNNs for speech recognition and speaker classification were trained using the second corpus. After the pretraining, VAEs were trained using the second corpus and evaluated using the first corpus. Note that the six speakers (i.e., the three males and the three females) were excluded from the training data. The speakers' some utterances were therefore used for estimating their speaker representations.

The performances of the following four DNNs were compared:

FFNN: Feed-Forward DNNs
VAE-SC: Conventional VAEs conditioned by speaker codes [38]
VAE-SC-PPG: Proposed VAEs conditioned by speaker codes and PPGs
VAE-DV-PPG: Proposed VAEs conditioned by d-vectors and PPGs

In the one-to-one VC evaluation, "VAE-*" were trained with a *completely non-parallel* speech corpus, while "FFNN" was trained with a *fully-parallel* speech corpus aligned with the DTW algorithm. Therefore, "FFNN" can be regarded as the ideal baseline of the

VC. The proposed "VAE-SC-PPG" and "VAE-DV-PPG" were also evaluated in many-to-many VC. In performing many-to-many VC with "VAE-SC-PPG," each target speaker's speaker representation was estimated using the speaker code adaptation method [118]. This adaptation method performs the BP algorithm to find a speaker representation that minimizes the MSE between target and generated speech parameters.

The STRAIGHT vocoder [47] was used for extracting the 0th-through-39th mel-cepstral coefficients, 10 band-aperiodicities, log F_0, and U/V at 5 ms steps. During the training phase, the mel-cepstral coefficients were normalized to have zero-mean and unit-variance. In the VC phase, the 1st-through-39th mel-cepstral coefficients and their dynamic features were converted by the DNNs. The input speaker's 0th mel-cepstral coefficients and band-aperiodicity were not converted. Input F_0 was linearly transformed using the F_0 statistics of the source and target speakers. The MLPG algorithm [61] was performed to generate static mel-cepstral coefficients considering their temporal dependencies.

All architectures for the DNNs and VAEs were Feed-Forward networks. The speech recognition DNNs used a set of 56 Japanese phonemes and predicted 56-dimensional PPGs frame by frame. The hidden layers of the DNNs had 4×1024-units with sigmoid non-linearity. The speaker classification DNNs predicted posterior probabilities of the speaker identity. Here, in addition to the 260 seen speakers, one discrete value representing an unvoiced region was attached to the speaker identity. The hidden layers of the speaker classification DNNs had 4×256-units with sigmoid non-linearity. Sixteen-dimensional d-vectors were extracted from the bottleneck layer of the DNNs. The optimization algorithm for pretraining the two DNNs was AdaGrad [98]. The learning rate for pretraining was set to 0.01. The pretraining was performed with 100 iterations. The encoder networks of the VAEs had two hidden layers with the ReLU [54] non-linearity. The first and second hidden layers had 256 and 128 units, respectively. The architecture for the decoder networks was symmetric about that for the encoder. The dimensionality of the VAE latent variables was 64. Feed-Forward DNNs ("FFNN") were trained using fully-parallel speech corpora including only source and target speakers. The hidden layers of the DNNs had 4×128 units with the ReLU non-linearity. The optimization algorithm for training the VAEs and "FFNN" was AdaGrad. The learning rate for the training was set to 0.01. All DNN-based acoustic models for VC were trained with 25 iterations.

5.3.2 Objective Evaluation

Mel-cepstral distortions (MCDs) between target and converted mel-cepstral coefficients were calculated. The frame lengths of these mel-cepstral coefficients were aligned using

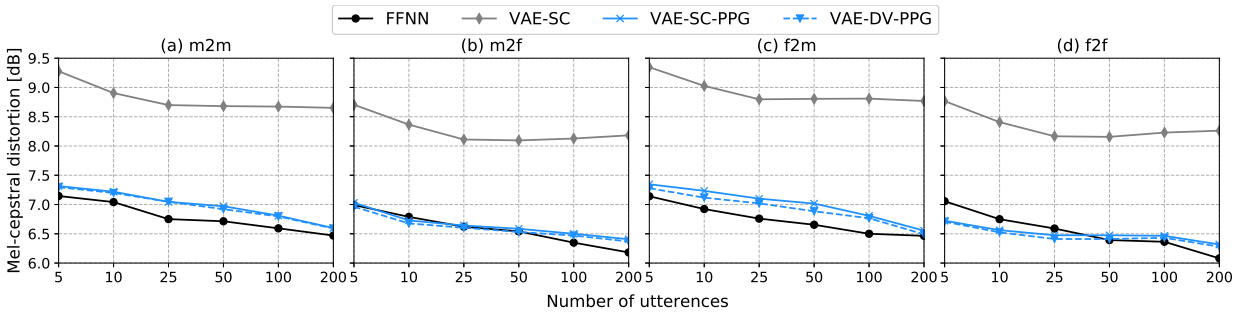

Fig. 5.9. MCDs of converted speech in one-to-one VC. Here, only "FFNN" was trained using fully-parallel speech corpora.

the DTW algorithm. The number of utterances for training DNNs in one-to-one VC or estimating speaker representations in many-to-many VC was changed. In one-to-one VC, the four DNNs were trained using 5, 10, 25, 50, 100, or 200 utterances. In many-to-many VC, speaker representations for the target speakers were estimated using the same numbers of utterances as used in one-to-one VC. The MCDs were averaged over all of the possible VC settings in each of the "m2m," "m2f," "f2m," and "f2f" conversion.

Figure 5.9 shows the evaluation results in one-to-one VC. The MCDs of the proposed *non-parallel* "VAE-SC-PPG" and "VAE-DV-PPG" significantly improve compared with those of the conventional "VAE-SC," and become closer to those of "FFNN" trained with *parallel* speech corpora. Moreover, the proposed methods even outperform "FFNN" when the number of the training utterances is small. One possible reason is misalignment by the DTW-based pre-processing; i.e., "FFNN" requires the DTW to align features, but the proposed methods do not. Also, the MCDs of "VAE-DV-PPG" are slightly lower than those of "VAE-SC-PPG," suggesting that the continuous speaker representations work better in VAE-based non-parallel VC.

Figure 5.10 shows the evaluation results of the proposed VAE-based non-parallel and many-to-many VC method. In all of the VC settings, the MCDs of "VAE-DV-PPG" are always lower than those of "VAE-SC-PPG," regardless of the number of utterances used for the speaker adaptation. These results indicate that using d-vectors is more effective than adapting speaker codes in VAE-based non-parallel and many-to-many VC.

5.3.3 Subjective Evaluation

Subjective evaluations on naturalness and speaker similarity of converted speech were conducted. The following six DNNs were compared: the two conventional DNNs ("FFNN" and "VAE-SC") and the two proposed DNNs ("VAE-SC-PPG" and "VAE-DV-PPG ") for

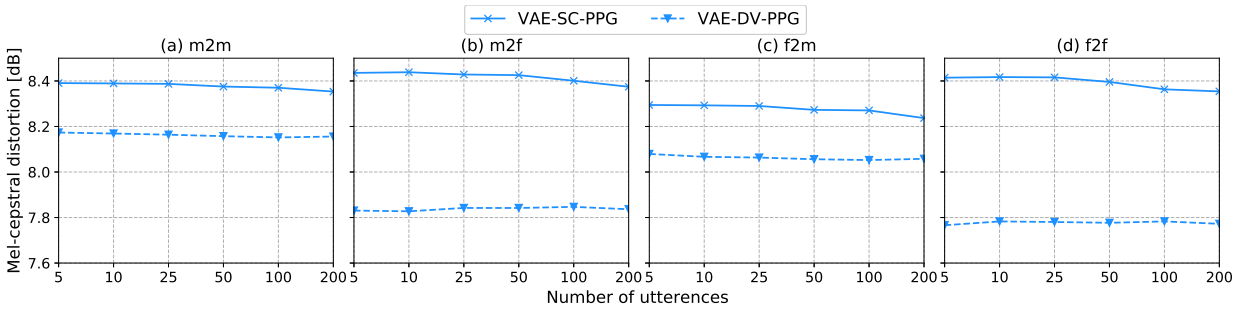

Fig. 5.10. MCDs of converted speech in VAE-based non-parallel and many-to-many VC

Table 5.1. Subjective evaluation results for naturalness of speech converted with different six DNNs and their 95% confidence intervals. **Bold** and underlined scores mean highest and lowest ones among five DNNs except for "FFNN," respectively

VC setting	DNNs	m2m	m2f	f2m	f2f
One-to-one	FFNN	3.77±0.13	3.30±0.13	3.68±0.14	3.53±0.12
	VAE-SC	1.09±0.05	1.12±0.06	1.07±0.04	1.10±0.05
	VAE-SC-PPG	**3.33±0.13**	**3.21±0.13**	3.29±0.12	**3.08±0.13**
	VAE-DV-PPG	**3.33±0.13**	**3.21±0.13**	**3.45±0.15**	3.04±0.13
Many-to-many	VAE-SC-PPG	2.94±0.11	2.88±0.12	2.94±0.12	2.87±0.13
	VAE-DV-PPG	2.73±0.11	2.75±0.12	2.85±0.13	2.69±0.11

both of the one-to-one and many-to-many VC settings. The fully-parallel 400 utterances of the source and target speakers were only used in "FFNN." The non-parallel 200 utterances were used in "VAE-*" for one-to-one VC. The target speakers' 100 utterances were used for the speaker representation estimation in VAE-based many-to-many VC. Five-point scale MOS tests were conducted for evaluating the naturalness. Speech samples generated by using each acoustic model were presented to listeners in random order. Similarly, five-point scale differential MOS (DMOS) tests were conducted for evaluating the speaker similarity. The reference samples were re-synthesized speech presented with corresponding converted speech. Fifty listeners participated in each of the evaluations for "m2m," "m2f," "f2m," and "f2f" conversion, using crowdsourced evaluation systems. Each listener evaluated the naturalness or speaker similarity of 30 converted speech samples randomly selected from all possible combinations of the test utterances and target speakers. The total number of listeners was 400. Similar to the objective evaluation in Section 5.3.2, the results of the MOS or DMOS tests were averaged over all of the possible VC settings.

Tables 5.1 and 5.2 shows the MOS and DMOS results, respectively. In this evaluation, only "FFNN" was trained with fully-parallel utterances, and the scores were referred to the ideal baseline. Focusing on the one-to-one VC results, the proposed methods ("VAE-*-PPG") achieve significantly higher scores than the conventional "VAE-SC" in

Table 5.2. Subjective evaluation results for speaker similarity of speech converted with different six DNNs and their 95% confidence intervals. **Bold** and underlined scores mean highest and lowest ones among five DNNs except for "FFNN," respectively

VC setting	DNNs	m2m	m2f	f2m	f2f
One-to-one	FFNN	3.49±0.16	3.20±0.14	3.13±0.14	2.99±0.15
	VAE-SC	1.28±0.10	1.21±0.08	1.24±0.09	1.27±0.09
	VAE-SC-PPG	**3.23±0.14**	**2.90±0.13**	**3.17±0.14**	**2.91±0.13**
	VAE-DV-PPG	3.13±0.14	2.87±0.13	3.04±0.13	2.84±0.14
Many-to-many	VAE-SC-PPG	2.08±0.12	1.92±0.12	2.14±0.12	2.09±0.12
	VAE-DV-PPG	2.31±0.12	1.94±0.11	2.33±0.11	2.10±0.12

terms of both naturalness and speaker similarity. These results demonstrate that the speech-recognition-derived phonetic latent variables successfully improve converted speech quality in the VAE-based non-parallel VC. Focusing on the many-to-many VC results, the MOS and DMOS of the proposed "VAE-*-PPG" are lower than those in one-to-one VC, although they are still superior to the conventional "VAE-SC." This is reasonable because the target speakers are unseen in many-to-many VC but are seen in one-to-one VC. These results suggest that the conventional VAE-based non-parallel VC can be extended to many-to-many VC using the proposed multi-speaker acoustic model. Regarding speaker representation estimation in many-to-many VC, the speaker code adaptation is effective to improve the naturalness, while the use of d-vectors is effective to improve the speaker similarity of converted speech.

5.3.4 Effects of Training Data and d-vector Dimensionality

The effects of hyperparameters in the proposed VAE-based non-parallel and many-to-many VC, i.e., the number of seen speakers and d-vector dimensionality, were investigated. The number of seen speakers was changed with three settings: 25 males and 25 females ("50spk"), 65 males and 65 females ("130spk"), and 130 males and 130 females ("260spk"). The d-vector dimensionality was changed with six settings: 1 ("1d"), 2 ("2d"), 4 ("4d"), 8 ("8d"), 16 ("16d"), and 32 ("32d"). In total, there were 18 hyperparameter settings to be investigated.

Evaluation of d-vector Dimensionality

The effect of d-vector dimensionality was investigated. Firstly, the number of seen speakers was fixed to 260, and speech samples converted by the proposed VC method with the six different settings for the d-vector dimensionality (i.e., "1–32d") were compared. Similar to Section 5.3.3, MOS and DMOS tests were conducted for the evaluation of naturalness and

Table 5.3. Subjective evaluation results for naturalness of speech converted by proposed VAE-based VC methods with different d-vector dimensionality and their 95% confidence intervals. **Bold** and underlined scores mean highest and lowest ones among six settings, respectively

	m2m	m2f	f2m	f2f
1d	2.60±0.12	2.63±0.13	2.67±0.14	2.80±0.14
2d	2.71±0.12	2.64±0.13	2.67±0.14	2.80±0.14
4d	2.80±0.13	2.58±0.13	2.84±0.14	2.75±0.14
8d	2.79±0.13	2.60±0.13	2.75±0.12	2.83±0.14
16d	2.85±0.13	2.61±0.14	**2.89±0.13**	**2.86±0.14**
32d	**2.94±0.14**	**2.64±0.14**	2.83±0.13	2.84±0.14

Table 5.4. Subjective evaluation results for speaker similarity of speech converted by proposed VAE-based VC methods with different d-vector dimensionality and their 95% confidence intervals. **Bold** and underlined scores mean highest and lowest ones among six settings, respectively

	m2m	m2f	f2m	f2f
1d	2.14±0.13	2.32±0.14	2.01±0.13	2.11±0.14
2d	2.18±0.14	2.26±0.13	2.01±0.13	2.12±0.13
4d	2.39±0.15	2.19±0.13	2.40±0.14	2.07±0.13
8d	2.44±0.14	2.42±0.14	**2.45±0.13**	2.18±0.13
16d	2.41±0.14	2.39±0.13	2.36±0.14	**2.25±0.14**
32d	**2.49±0.14**	**2.47±0.14**	2.43±0.15	2.22±0.13

speaker similarity of converted speech, respectively. Fifty listeners participated in each of the evaluation, using crowdsourced evaluation systems. The total number of listeners was 400.

Tables 5.3 and 5.4 show the MOS and DMOS results, respectively. From the results shown in Table 5.4, the speaker similarity significantly degrades when the d-vector dimensionality is set to smaller than eight. These results are consistent with the speaker verification performance described in Appendix C, and suggest that the use of inaccurate d-vectors to distinguish speaker individuality degrades speaker similarity of converted speech. Henceforth, "1d," "2d," and "4d" were omitted from the following pairwise comparisons.

Secondly, the number of seen speakers was fixed to 50, 130, or 260, and speech samples converted by the proposed VC method with the three different d-vector dimensionality settings (i.e., "8d," "16d," or "32d") were compared. Preference AB tests for the naturalness and preference XAB tests for the speaker similarity were conducted to find optimal settings of the d-vector dimensionality. Twenty-five listeners participated in each of the

Table 5.5. Subjective evaluation results for naturalness of converted speech with fixed numbers of seen speakers and varied d-vector dimensionality. **Bold** indicates more preferred method with p-value < 0.05

spk		8d vs. 16d	16d vs. 32d	8d vs. 32d
50	m2m	0.315 - **0.685**	**0.577** - 0.423	0.396 - **0.604**
	m2f	0.442 - **0.558**	0.444 - **0.556**	0.337 - **0.663**
	f2m	0.337 - **0.663**	0.524 - 0.476	0.384 - **0.616**
	f2f	0.368 - **0.632**	0.464 - 0.536	0.304 - **0.696**
130	m2m	0.468 - 0.532	**0.600** - 0.400	0.428 - **0.572**
	m2f	0.492 - 0.508	**0.664** - 0.336	0.538 - 0.462
	f2m	0.396 - **0.604**	**0.664** - 0.336	0.415 - **0.585**
	f2f	0.512 - 0.488	**0.692** - 0.308	**0.568** - 0.432
260	m2m	0.512 - 0.488	0.468 - 0.532	0.532 - 0.468
	m2f	0.500 - 0.500	0.516 - 0.484	0.532 - 0.468
	f2m	0.472 - 0.528	0.492 - 0.502	**0.552** - 0.448
	f2f	0.524 - 0.476	**0.556** - 0.444	0.504 - 0.496

evaluations, using crowdsourced evaluation systems. The total number of listeners was 1,800. Each listener evaluated 10 samples.

Tables 5.5 and 5.6 show evaluation results for converted speech naturalness and speaker similarity, respectively. Two noteworthy points are found from the results. When the number of seen speaker is small, the use of the higher d-vector dimensionality (i.e., "16d" and "32d") significantly improves both the naturalness and speaker similarity in most cases. In contrast, these tendencies are not observed when the largest number of seen speakers (i.e., "260spk") is used, and "8d" even achieves higher speaker similarity than "32d." These results suggest that a trade-off between the d-vector dimensionality and number of seen speakers exists in the proposed method. In other words, the d-vector dimensionality should be large enough to deal with unseen speakers in the VC process when the small number of seen speakers is used.

Evaluation of Number of Seen Speakers

Thirdly, the d-vector dimensionality was fixed to the best settings referring to preference scores shown in Tables 5.5 and 5.6, and speech samples converted by the proposed VC method with the three different number of seen speakers (i.e., "50spk," "130spk," or "260spk") were compared. Tables 5.7 and 5.8 show the subjective evaluation results for naturalness and speaker similarity, respectively. The increase of the number of seen speakers tends to improve both the naturalness and speaker similarity. The RMSEs between two PPGs predicted from source and target speakers' utterances were calculated

Table 5.6. Subjective evaluation results for speaker similarity of converted speech with fixed numbers of seen speakers and varied d-vector dimensionality. **Bold** indicates more preferred method with p-value < 0.05

spk		8d vs. 16d	16d vs. 32d	8d vs. 32d
50	m2m	0.450 - **0.550**	0.485 - 0.515	0.528 - 0.472
	m2f	0.472 - 0.528	0.472 - 0.528	0.476 - 0.524
	f2m	0.420 - **0.580**	0.468 - 0.532	0.412 - **0.588**
	f2f	0.462 - 0.538	0.496 - 0.504	0.438 - **0.562**
130	m2m	0.428 - **0.572**	0.508 - 0.492	0.454 - **0.546**
	m2f	**0.548** - 0.452	0.480 - 0.520	0.528 - 0.472
	f2m	0.484 - 0.516	0.504 - 0.496	0.500 - 0.500
	f2f	**0.588** - 0.412	0.476 - 0.524	0.560 - 0.440
260	m2m	0.452 - **0.548**	0.527 - 0.473	0.516 - 0.484
	m2f	0.488 - 0.512	0.536 - 0.464	0.460 - 0.540
	f2m	0.464 - 0.536	0.540 - 0.460	0.432 - **0.568**
	f2f	0.508 - 0.492	0.464 - 0.536	**0.569** - 0.431

Table 5.7. Subjective evaluation results for naturalness of converted speech with varied numbers of seen speakers and best d-vector dimensionality for each setting. Here, d-vector dimensionality was "32d" for "50spk," "16d" for "130spk," and "8d" for "260spk," respectively. **Bold** indicates more preferred method with p-value < 0.05

	50spk vs. 130spk	130spk vs. 260spk	50spk vs. 260spk
m2m	0.316 - **0.684**	0.460 - 0.540	0.240 - **0.760**
m2f	0.476 - 0.524	0.408 - **0.592**	0.440 - **0.560**
f2m	0.364 - **0.636**	0.438 - **0.562**	0.324 - **0.676**
f2f	0.496 - 0.504	0.492 - 0.508	0.473 - 0.527

to investigate the reason. The DTW algorithm was performed to align PPGs' frame lengths. Table 5.9 shows the results. The RMSEs tend to decrease in proportion to the number of seen speakers, which may account for the improvement in converted speech quality.

Visualization of Latent Variables

Finally, scatter plots of the d-vectors and the VAE latent variables were made using principal component analysis (PCA) to explore the latent spaces learned by DNNs. In the following visualization, the d-vectors or the VAE latent variables were averaged over 50 utterances of each speaker. Each point in the scatter plots corresponds to each speaker, and male and female speakers are colored with blue and red, respectively. The six speakers used in the evaluation are marked with the texts "M*" (male) or "F*" (female).

Table 5.8. Subjective evaluation results for speaker similarity of converted speech with varied numbers of seen speakers and the best d-vector dimensionality for each setting. Here, d-vector dimensionality was "32d" for "50spk," "16d" for "130spk," and "8d" for "260spk," respectively. **Bold** indicates more preferred method with p-value < 0.05

	50spk vs. 130spk	130spk vs. 260spk	50spk vs. 260spk
m2m	0.436 - **0.564**	0.476 - 0.524	0.285 - **0.715**
m2f	0.496 - 0.504	0.540 - 0.460	0.416 - **0.584**
f2m	0.408 - **0.592**	0.427 - **0.573**	0.360 - **0.640**
f2f	0.462 - 0.538	0.532 - 0.468	0.448 - **0.552**

Table 5.9. RMSEs of PPGs predicted from source and target speakers. These results were averaged over all evaluation data. **Bold** indicates lowest RMSE in each row

	50spk	130spk	260spk
m2m	0.0564	**0.0534**	0.0539
m2f	0.0566	0.0542	**0.0541**
f2m	0.0566	0.0542	**0.0541**
f2f	0.0563	0.0541	**0.0539**

Figure 5.11 shows the visualization of d-vectors. The use of larger d-vector dimensionality tends to widen the distribution of speakers, and the six speakers ("M*" and "F*") shown in Fig. 5.11(c)(3) are clearly differentiated in the speaker space. However, such a speaker space does not always contribute to the improvement of converted speech quality as shown in Table 5.5. These results indicate that the densely distributed speaker space may be suited to the proposed VAE-based VC, rather that the space constructed for the better speaker verification performance.

Figure 5.12 shows the visualization of VAE latent variables. The "CSM" in this plot means the class separability measure calculated as the trace of the between-class scatter matrix divided by that of the within-class scatter matrix [119]. Since the CSM values were calculated regarding speakers as classes, the numerator and denominator of the CSM definition should represent inter-speaker variation and inter-phonetics variation, respectively. The CSM values shown in Fig. 5.12 are always greater than 1.0, indicating that the inter-speaker variation is dominant in the latent variables. In fact, some of the six test speakers tend to position out of the distribution of the latent variables when the smaller number of seen speakers is used (e.g., "M1" and "M3" in Fig. 5.12(c)(1)). These results suggest that the VAE latent variables can quantify how well the VAEs fit the speaker individuality, rather than phonetic contents suggested in [38].

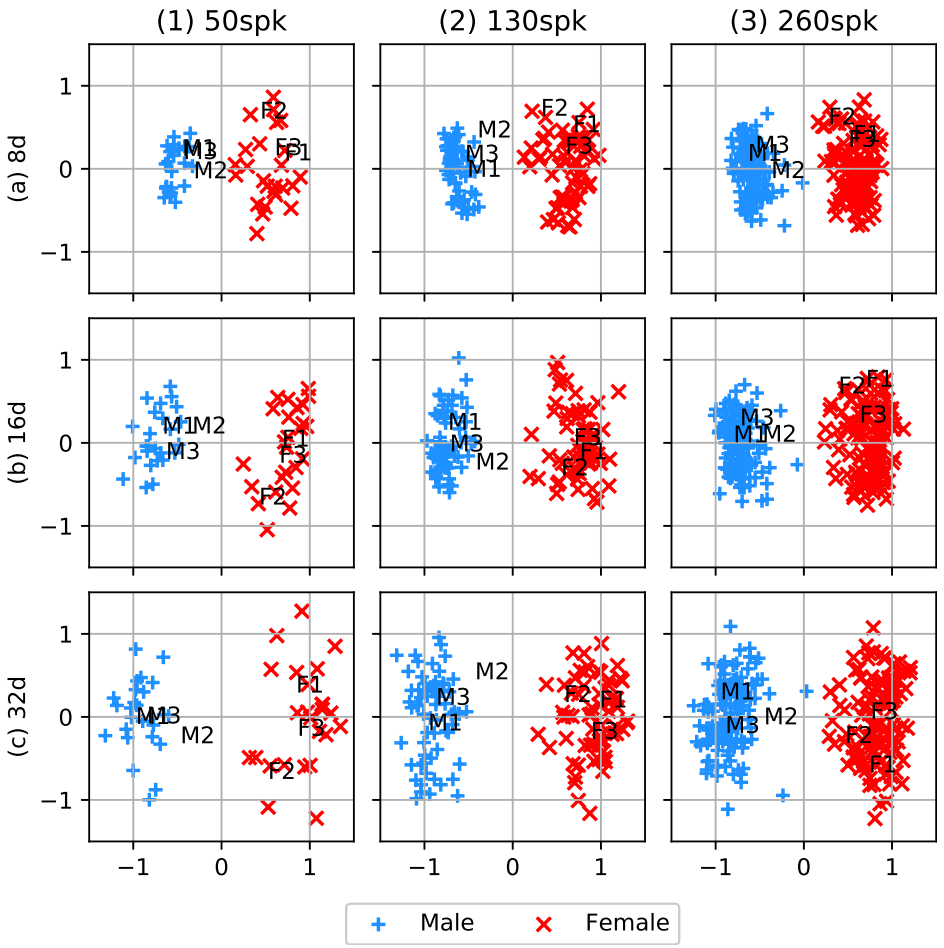

Fig. 5.11. Scatter plots of d-vectors compressed with PCA. Each point denotes representation of one speaker, and three male and three female speakers used in evaluation were marked with texts "M*" and "F*" in this plot, respectively. To place males' cluster in left side and females' cluster in right side, some figures were rotated 180 degrees.

5.4 Summary

This chapter proposed the VAE-based multi-speaker speech synthesis method integrating speech recognition and speaker classification to synthesize versatile speakers' high-quality voices. This method introduces a phonetic latent variable predicted by a DNN-based speech recognition model to improve synthetic speech quality. It also uses a speaker latent variable extracted from a DNN-based speaker classification model to increase the speaker diversity of synthetic speech. VAE-based non-parallel and many-to-many VC that can reproduce and transform arbitrary speakers' voice characteristics is established as a result. Experimental results demonstrated that the phonetic latent variables significantly

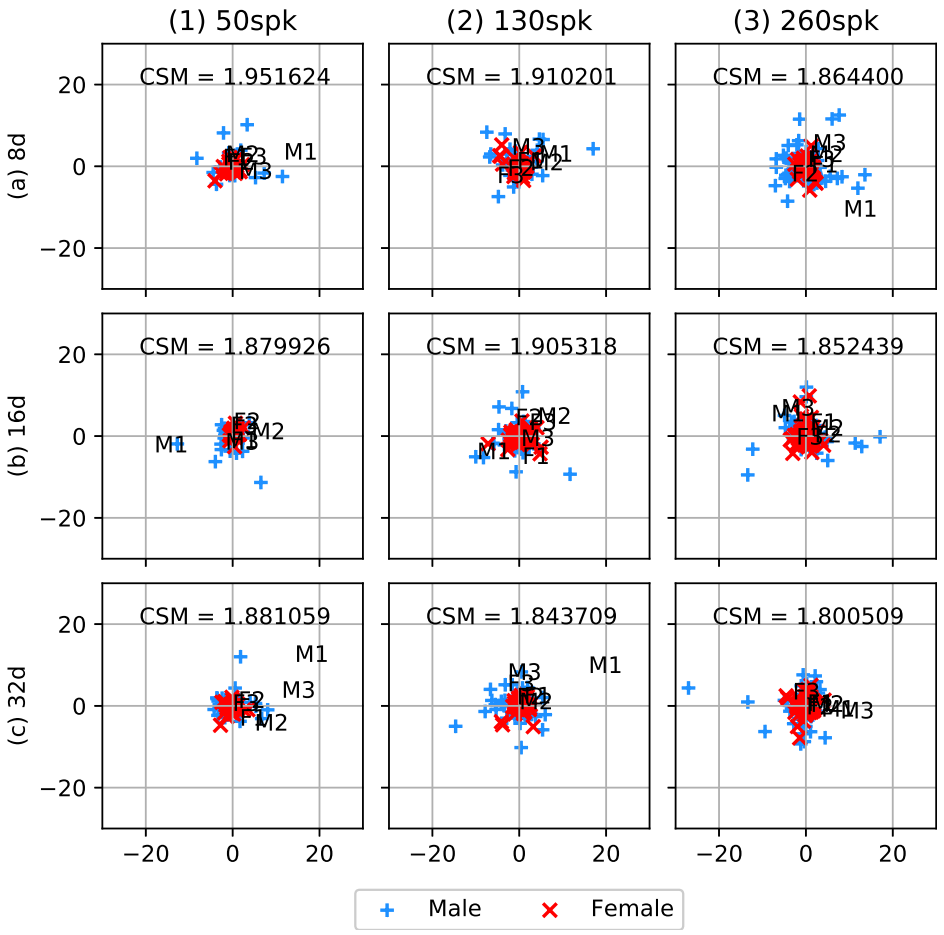

Fig. 5.12. Scatter plots of VAE latent variables compressed with PCA. Each point denotes latent variable averaged over 50 utterances of each speaker, and three male and three female speakers used in evaluation were marked with texts "M*" and "F*" in this plot, respectively. "CSM" in this plot means class separability measure calculated as trace of between-class scatter matrix divided by that of within-class scatter matrix.

improved both naturalness and speaker similarity of the converted speech. The results also indicated that the speaker latent variables were effective speaker representations for the VAE-based non-parallel and many-to-many VC. The effects of hyperparameters of the proposed method were investigated. The investigation results indicated that 1) a large d-vector dimensionality that gives the better speaker verification performance did not necessarily improve converted speech quality, and 2) a large number of seen speakers tended to improve converted speech quality.

Chapter 6

Perceptual-similarity-aware Deep Speaker Representation Learning

6.1 Introduction

This chapter proposes novel algorithms for learning speaker representations using DNNs, i.e., deep speaker representation learning, considering perceptual similarity among speakers. As described in Chapter 5, d-vectors, speaker representations learned by speaker classification DNNs, can be used for controlling speaker individuality in DNN-based multi-speaker acoustic modeling. However, they do not consider the perceptual similarity among speakers, resulting in less interpretability for the speaker individuality control. Also, they can even worsen synthetic speech quality when the speaker is unseen [120]. The proposed algorithms incorporate human listeners to the speaker representation learning framework, on the basis of human computation [121]. Figure 6.1 illustrates a conceptual diagram of the proposed deep speaker representation learning algorithms. These algorithms first conduct large-scale scoring of perceptual speaker-pair similarity to obtain a *perceptual speaker similarity matrix*, then train DNNs for speaker representation learning (i.e., speaker encoder) to minimize a loss function defined by the similarity matrix. This chapter presents three algorithms with different representations of the similarity matrix: a set of similarity vectors, the Gram matrix, and a graph. Similarity *vector* embedding regards the similarity matrix as a set of similarity vectors and trains a speaker encoder to predict a similarity vector from a speaker representation. Similarity *matrix* embedding directly utilizes the

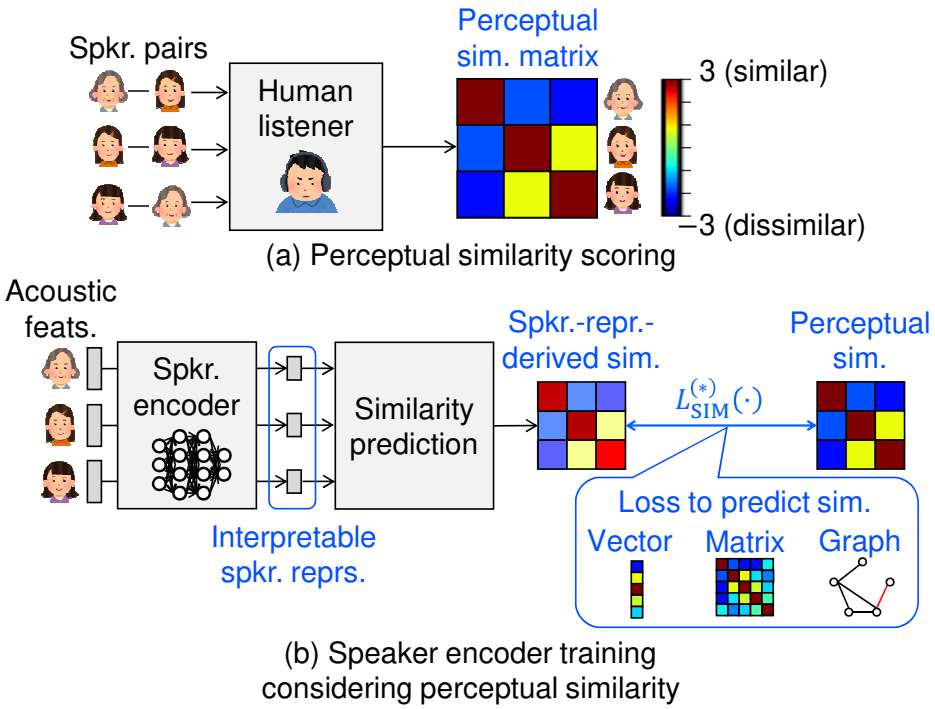

Fig. 6.1. Conceptual diagram of proposed method in Chapter 6

whole similarity matrix as the target to be predicted by a speaker encoder and trains the encoder to minimize the Frobenius norm between the Gram matrix of a set of speaker representations (i.e., speaker similarity matrix derived from the representations) and the perceptual speaker similarity matrix. Similarity *graph* embedding defines a graph that represents perceptual speaker similarity and trains a speaker encoder to predict a link of the graph from a pair of speaker representations. This chapter also proposes an active learning algorithm to reduce the number of scoring times that quadratically increases with that of seen speakers. This algorithm iterates the perceptual speaker-pair similarity scoring and speaker encoder training. Queries in this algorithm are generated from the sequentially-trained speaker encoder for prioritizing unscored speaker-pairs to be scored next. Figure 6.2 illustrates the relation between conventional and proposed methods. In experimental evaluations, large-scale scoring is first conducted, then the proposed speaker representation learning algorithms and active learning algorithm are performed.

This chapter is organized as follows (see also Fig. 6.3). Section 6.2 describes the three components in the proposed method: scoring perceptual speaker-pair similarity, deep speaker representation learning considering the similarity, and active learning. Section 6.3 presents experimental evaluations. Section 6.4 summarizes this chapter.

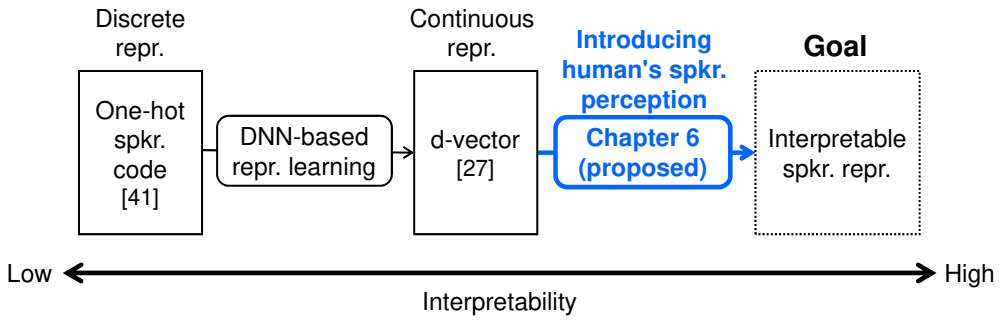

Fig. 6.2. Relation between conventional and proposed methods in Chapter 6

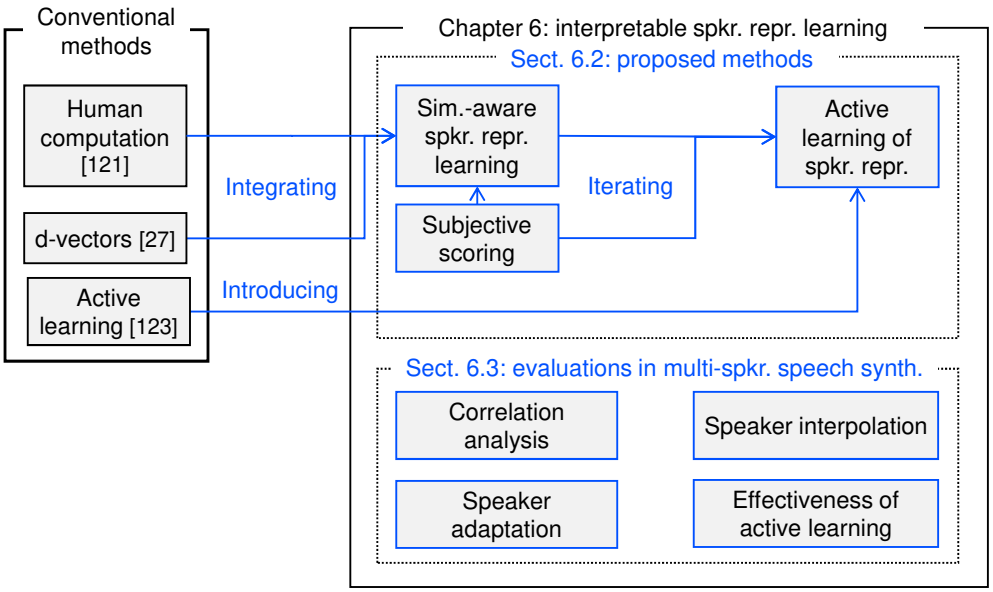

Fig. 6.3. Overview of Chapter 6

6.2 Deep Speaker Representation Learning Considering Perceptual Speaker-pair Similarity

The introduction of human speaker-similarity perception into deep speaker representation learning can improve controllability of a DNN-based multi-speaker acoustic model and the adaptability of the model to unseen speakers. In this section, a speaker similarity matrix obtained by perceptual scoring is first introduced. Speaker representation learning algorithms using the matrix are then described. An active learning algorithm to reduce the scoring and training costs is finally presented.

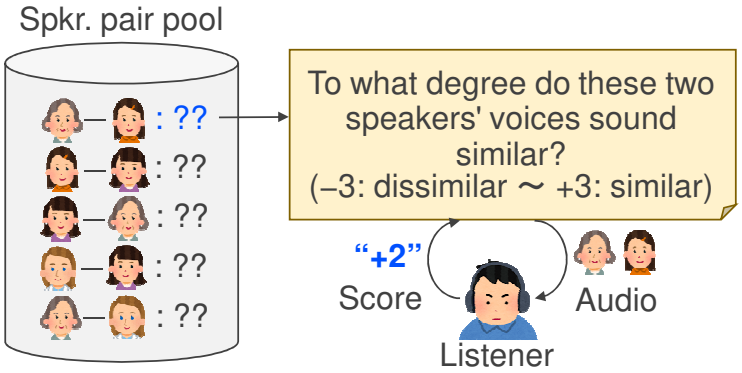

Fig. 6.4. Perceptual scoring of speaker-pair similarity. Speaker-pair pool stores speaker pairs to be scored. Listener is asked to score perceptual similarity of two presented speakers' voices as integer between $-v$ and v. In this figure, $v = 3$.

6.2.1 Perceptual Speaker Similarity Matrix

A perceptual speaker similarity matrix that represents the pairwise speaker similarity perceived by listeners is defined. Let $\mathbf{S} = [\boldsymbol{s}_1, \cdots, \boldsymbol{s}_i, \cdots, \boldsymbol{s}_{N_\mathrm{s}}]$ be an N_s-by-N_s symmetric similarity matrix and $\boldsymbol{s}_i = [s_{i,1}, \cdots, s_{i,j}, \cdots, s_{i,N_\mathrm{s}}]^\top$ be an N_s-dimensional similarity vector of the ith speaker. Each element $s_{i,j}$ takes a value between $-v$ and v that represents the perceptual similarity of the ith and jth speakers. The element $s_{i,j}$ is defined as the average score of perceptual scoring that asks listeners "To what degree do the ith speaker's voice and the jth speaker's sound similar? Please answer the degree of similarity as a value between $-v$ and v." The diagonal elements $s_{i,i}$, i.e., intra-speaker perceptual similarity, are assumed to take the maximum value v. Figure 6.4 illustrates the perceptual scoring process. Figures 6.5(a) and (b) show a perceptual speaker similarity matrix of 153 female Japanese speakers and its sub-matrix, respectively. Please see Section 6.3.1 for details of the perceptual scoring and Section 6.3.2 for the analysis results of the scores.

6.2.2 Perceptual-similarity-aware Deep Speaker Representation Learning Algorithms

This section presents three algorithms for learning similarity-aware speaker representations with different usage of the perceptual speaker similarity matrix: a set of similarity vectors, the Gram matrix, and a graph.

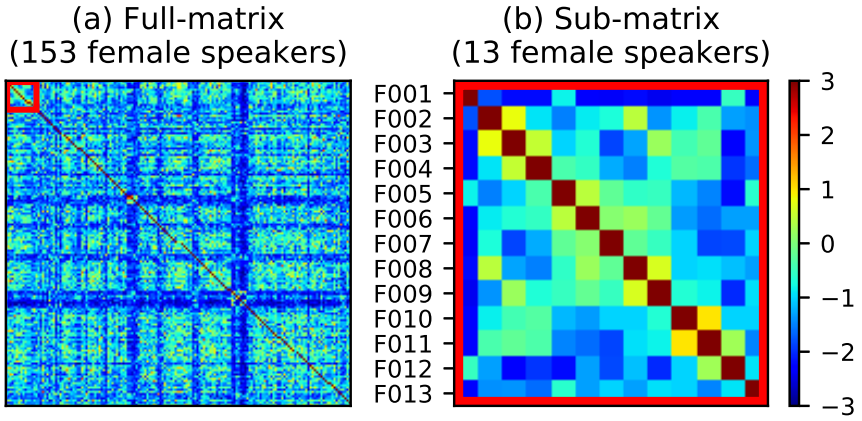

Fig. 6.5. (a) Perceptual speaker similarity matrix of 153 female Japanese speakers obtained by large-scale perceptual scoring and (b) its sub-matrix

Similarity Vector Embedding

The first algorithm uses a speaker similarity vector as the target to be predicted by a speaker encoder. A loss function for the training is defined as follows:

$$L_{\mathrm{SIM}}^{(\mathrm{vec})}\left(\boldsymbol{s}, \hat{\boldsymbol{s}}\right) = \frac{1}{N_{\mathrm{s}}}\left(\hat{\boldsymbol{s}} - \boldsymbol{s}\right)^{\top}\left(\hat{\boldsymbol{s}} - \boldsymbol{s}\right), \tag{6.1}$$

where $\boldsymbol{s} \in \mathbf{S}$ and $\hat{\boldsymbol{s}}$ denote a target similarity vector and output vector of the DNNs, respectively. This algorithm can be regarded as speaker classification based on continuous-valued speaker identity considering perceptual speaker similarity. Figure 6.6(a) shows the computation procedure of $L_{\mathrm{SIM}}^{(\mathrm{vec})}(\cdot)$.

Similarity Matrix Embedding

The second algorithm directly uses a perceptual speaker similarity matrix as a constraint on coordinates of speaker representations in the speaker space. Let $\mathbf{D} = [\boldsymbol{d}_1, \cdots, \boldsymbol{d}_i, \cdots, \boldsymbol{d}_{N_{\mathrm{s}}}]$ be an N_{d}-by-N_{s} matrix including all seen speakers' representations. A loss function for the training is defined as follows:

$$L_{\mathrm{SIM}}^{(\mathrm{mat})}\left(\mathbf{D}, \mathbf{S}\right) = \frac{2}{\left\|\mathbf{1}_{N_{\mathrm{s}}} - \mathbf{I}_{N_{\mathrm{s}}}\right\|_F^2}\left\|\widetilde{\mathbf{K}}_{\mathbf{D}} - \widetilde{\mathbf{S}}\right\|_F^2, \tag{6.2}$$

$$\widetilde{\mathbf{K}}_{\mathbf{D}} = \mathbf{K}_{\mathbf{D}} - \left(\mathbf{K}_{\mathbf{D}} \circ \mathbf{I}_{N_{\mathrm{s}}}\right), \tag{6.3}$$

$$\widetilde{\mathbf{S}} = \mathbf{S} - v\mathbf{I}_{N_{\mathrm{s}}}, \tag{6.4}$$

where $\|\cdot\|_F$, $\mathbf{1}_{N_{\mathrm{s}}}$, and $\mathbf{I}_{N_{\mathrm{s}}}$ denote the Frobenius norm of a given matrix, an N_{s}-by-N_{s} matrix whose components are all 1, and the N_{s}-by-N_{s} identity matrix, respectively. The

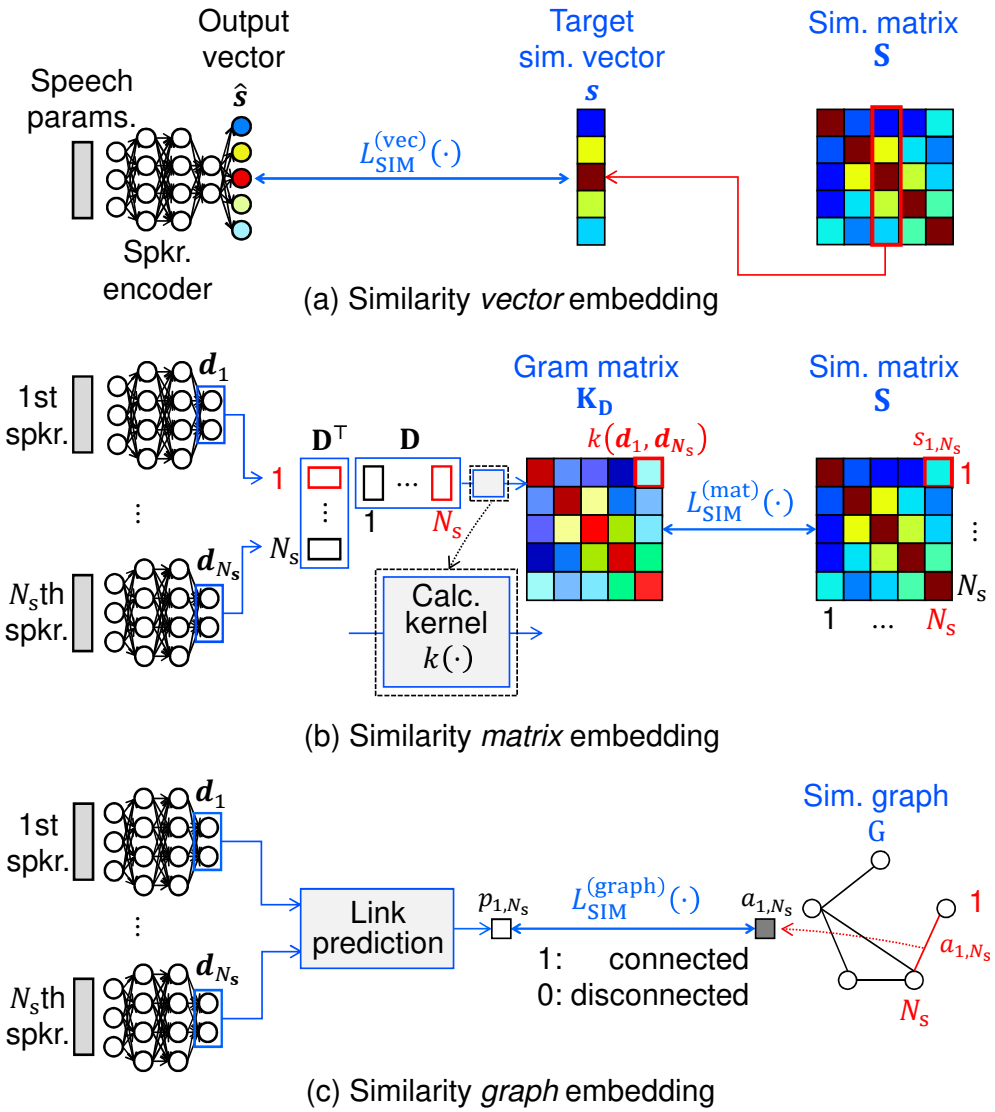

Fig. 6.6. Calculation of loss functions in proposed algorithms based on (a) similarity vector embedding, (b) similarity matrix embedding, and (c) similarity graph embedding

normalization coefficient $2/\|\mathbf{1}_{N_s} - \mathbf{I}_{N_s}\|_F^2$ corresponds to the degrees of freedom of the matrix $\widetilde{\mathbf{K}}_{\mathbf{D}} - \widetilde{\mathbf{S}}$. The Gram matrix of a set of speaker representations $\mathbf{K}_{\mathbf{D}}$ is defined as:

$$\mathbf{K}_{\mathbf{D}} = \begin{bmatrix} k(\boldsymbol{d}_1, \boldsymbol{d}_1) & \cdots & k(\boldsymbol{d}_1, \boldsymbol{d}_{N_s}) \\ \vdots & \ddots & \vdots \\ k(\boldsymbol{d}_{N_s}, \boldsymbol{d}_1) & \cdots & k(\boldsymbol{d}_{N_s}, \boldsymbol{d}_{N_s}) \end{bmatrix}, \tag{6.5}$$

where $k(\boldsymbol{d}_i, \boldsymbol{d}_j)$ is a kernel function of a pair of speaker representations $\boldsymbol{d}_i$ and $\boldsymbol{d}_j$, i.e., speaker similarity derived from the speaker representations. This proposed algorithm therefore makes the speaker-representation-derived similarity closer to the perceptual sim-

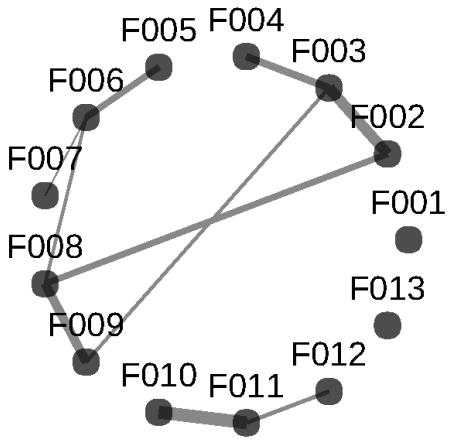

Fig. 6.7. Speaker similarity graph defined by similarity matrix shown in Fig. 6.5(b). Each node represents one speaker, and links connect perceptually similar pairs (i.e., $s_{i,j} > 0$). Wider links indicate more similar speaker pairs.

ilarity. Figure 6.6(b) shows the computation procedure of $L_{\mathrm{SIM}}^{(\mathrm{mat})}(\cdot)$.

Similarity Graph Embedding

The third algorithm learns relations among speakers defined by a subjective speaker similarity matrix. Let G be a speaker similarity graph defined by the matrix $\mathbf{S}$. Each node of the graph G represents the speaker identity of one speaker, and a link connects a pair of similar speakers, as shown in Fig. 6.7. An N_{s}-by-N_{s} adjacency matrix $\mathbf{A}$ is defined to determine the existence of the links based on elements of the similarity matrix $\mathbf{S}$. A loss function for the training is defined as follows:

$$L_{\mathrm{SIM}}^{(\mathrm{graph})}(\mathbf{D}, \mathbf{A}) = -\sum_{i,j=1, i\neq j}^{N_{\mathrm{s}}} a_{i,j} \log p_{i,j} - \sum_{i,j=1, i\neq j}^{N_{\mathrm{s}}} (1 - a_{i,j}) \log (1 - p_{i,j}), \tag{6.6}$$

where $a_{i,j}$ and $p_{i,j}$ denote an element of the adjacency matrix and link probability, respectively. The link probability is defined as $p_{i,j} = \exp(-||\boldsymbol{d}_i - \boldsymbol{d}_j||_2^2)$ referring to [122]. Figure 6.6(c) shows the computation procedure of $L_{\mathrm{SIM}}^{(\mathrm{graph})}(\cdot)$.

6.2.3 Active Learning of Perceptual-similarity-aware Speaker Representations

The proposed speaker representation learning algorithms require the perceptual scoring of speaker similarity, and the number of scoring times quadratically increases with that of seen speakers N_{s} as well as the training time. This section proposes an active learning

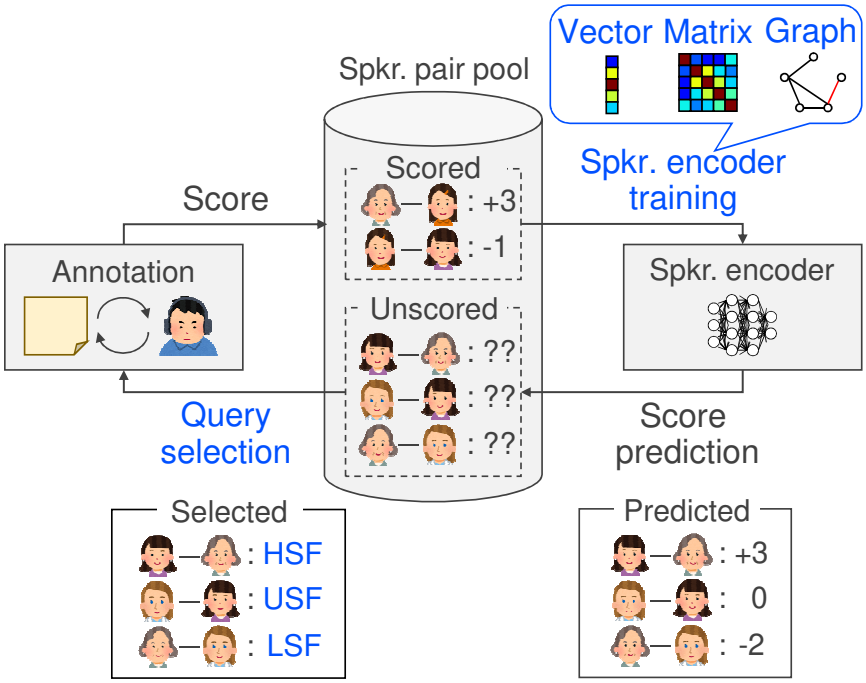

Fig. 6.8. Active learning of perceptual-similarity-aware speaker representations

algorithm to reduce the scoring and training costs. Active learning [123] is a general framework to sequentially train a machine learning model with a small labeled dataset and large unlabeled one. It iterates 1) model training with the labeled dataset and 2) query selection to increase labeled data.

Figure 6.8 shows the active learning of the proposed deep speaker representation learning. In the active learning, the ${}_{N_\mathrm{s}}\mathrm{C}_2$ seen speaker pairs are divided into two subsets: 1) scored pairs $\mathcal{D}_\mathrm{s}$ and 2) the remaining unscored ones $\mathcal{D}_\mathrm{u}$. The similarity scores of speaker pairs in $\mathcal{D}_\mathrm{u}$ are unobserved initially.

Speaker Encoder Training Using Scored Pairs

A speaker encoder is trained using scored pairs $\mathcal{D}_\mathrm{s}$ to learn the perceptual similarity among them. The loss function for the training is any of the proposed similarity vector, matrix, or graph embedding algorithms, i.e., Eqs. (6.1), (6.2), or (6.6).

Query Selection from Unscored Pairs

The trained speaker encoder first predicts queries (i.e., tentative similarity scores of unscored pairs $\mathcal{D}_\mathrm{u}$) that indicate which of the pairs should be scored preferentially. Then, an oracle (e.g., a human annotator) annotates scores to speaker pairs with higher priority. A query strategy has an important role in the query selection since it determines

the priority of scoring. Three query strategies are investigated in this chapter: 1) lower-similarity first (LSF) that selects a speaker pair whose predicted similarity is closer to $-v$, 2) higher-similarity first (HSF) that corresponds to the inverse version of the LSF, and 3) uncertain-similarity first (USF) that selects a speaker pair whose predicted similarity is closer to 0.

6.2.4 Discussion

Regarding prior work, Tachibana et al. [124] and Ohta et al. [125] proposed controllable statistical speech synthesis in the HMM and GMM era. They modeled a single speaker's voice characteristics with a pair of subjective impression words, e.g., "warm – cold" and "clear – hoarse," as latent variables of the HMMs and GMMs. The proposed speaker representation learning algorithms extend these ideas to make DNNs learn the *pairwise* speakers' perceptual similarity, rather than the conventional *pointwise* speaker's voice impression. Furthermore, one can model the relation between a speaker's intention and listener's perception (e.g., difference in emotion perception [126]) by using the algorithms.

The similarity matrix embedding in Section 6.2.2 can directly learn relations among speakers as the Gram matrix. One can choose an arbitrary kernel function to construct the speaker space. When the inner product is used as the kernel function, Eq. (6.2) is equivalent to deep clustering [127] (except for the diagonal component subtraction). Not only such a simple kernel but also a more complicated one can be utilized.

The similarity graph embedding in Section 6.2.2 introduces knowledge of graph embedding [128] to deep speaker representation learning. Therefore, one can further incorporate graph signal processing [129] and graph neural networks [130] to this algorithm for better modeling.

The proposed active learning algorithm in Section 6.2.3 can be regarded as human-in-the-loop (HITL) learning [131] of speaker representations considering human speech perception. From this viewpoint, one can extend the HITL learning framework to speech synthesis that considers a human listener's speech quality assessment for acoustic modeling (e.g., the GAN training incorporating a human-based discriminator [132]).

6.3 Experimental Evaluation

6.3.1 Experimental Conditions

Conditions for Large-scale Perceptual Scoring

Large-scale perceptual scoring was conducted to obtain the similarity matrix $\mathbf{S}$ using the JNAS corpus [133] that includes 153 Japanese female speakers. Each speaker utters at least 150 reading-style utterances (totaling about 44 hours). Five non-parallel utterances per speaker were used for scoring text-independent perceptual similarity among the speakers. Each listener scored the perceptual similarity of 34 randomly-selected speaker pairs extracted from all of the 11,628 possible different speaker pairs with an integer between -3 (very dissimilar) and $+3$ (very similar). Listeners were recruited using Lancers[*1], a well-known crowdsourcing platform in Japan. At least 10 different listeners scored the similarity of one of the 11,628 speaker pairs. The total numbers of listeners and answers were 4,060 and 138,040, respectively.

Conditions for Deep Speaker Representation Learning

The JNAS corpus was used for training a DNN-based speaker encoder. The 13 speakers shown in Fig. 6.5(b) (from "F001" to "F013") were assumed to be unseen during the training. In the training, 90% of the remaining 140 seen speakers' utterances were used. The number of utterances per speaker was balanced. In the evaluation, the unseen speakers' 50 utterances and the seen speakers' remaining ones were used. The five utterances used for the perceptual scoring were omitted from both the training and evaluation data.

During the training, each element in the similarity matrix was normalized to be in $[-1, +1]$ for the similarity vector or matrix embedding and in [0, 1] for the similarity graph embedding. Accordingly, the sigmoid kernel $k(\boldsymbol{d}_i, \boldsymbol{d}_j) = \tanh(\boldsymbol{d}_i^\top \boldsymbol{d}_j)$ was used for the Gram matrix calculation in Eq. (6.5). In the similarity graph embedding, the adjacency matrix was defined using the normalized similarity matrix that represents the likelihood of a link existence as a value between [0, 1].

The DNN architecture for the speaker encoder was a Feed-Forward network that included four hidden layers with the tanh activation function. The numbers of hidden units at the first-through-third layers and the fourth layer for the speaker representation extraction were 256 and 8, respectively. In the d-vector learning (Section 5.2.2) and the

[*1] `https://www.lancers.jp`

similarity vector embedding, an output layer with 140 units was prepared. The output activation function was the softmax for the former and the tanh for the latter. The input of the speaker encoder was a joint vector of the 1st-through-39th mel-cepstral coefficients and their dynamic features. The STRAIGHT vocoder [47] was used to extract the mel-cepstral coefficients. The mel-cepstral coefficients were normalized to have zero-mean and unit-variance during the training. The optimization algorithm was AdaGrad [98]. The learning rate was set to 0.01. The number of iterations for the training was 100.

Conditions for Multi-speaker Statistical Speech Synthesis

A VAE-based multi-speaker acoustic model (described in Chapter 5) was trained. The DNN architecture for the speech recognition model was a Feed-Forward network that included four hidden layers with the tanh activation function. The number of hidden units was 1,024. The recognition model was trained to predict 43-dimensional Japanese PPGs [111] from the same input vector as the speaker encoder. At least 50 utterances for each of the 140 seen speakers were used for the training. The number of iterations for the training was 100. The DNN architecture for the VAEs was a Feed-Forward network that consisted of encoder and decoder networks. The encoder had two hidden layers with the ReLU [54] activation function and extracted the 64-dimensional latent variables from a joint vector of the static-dynamic mel-cepstral coefficients and PPGs. The first and second hidden layers had 256 and 128 units, respectively. The decoder reconstructed the input static-dynamic mel-cepstral coefficients from a joint vector of the VAE latent variables, PPGs, and 8-dimensional speaker representations. The DNN architecture for the decoder was symmetric about that for the encoder. The VAEs were trained to maximize the variational lower bound of the log likelihood [37] with 25 iterations using the same training data as that used in the speaker encoder training. The optimization algorithm for the speech recognition model and VAEs was AdaGrad. The learning rate was set to 0.01. The MLPG algorithm [61] was used to generate static mel-cepstral coefficients considering their temporal dependencies. The generated mel-cepstral coefficients and original speech's excitation parameters (i.e., F_0 and five band-aperiodicity [44, 96]) were used for speech waveform synthesis using the STRAIGHT vocoder [47].

Conditions for Active Learning

In active learning, the 140 seen speakers were divided into two groups (the first 70 speakers and the remaining), and speaker similarity scores among the different groups were assumed to be unobserved as shown in Fig. 6.9(b). The active learning alternatively performed 1) the speaker encoder training using observed similarity scores with one iteration and 2) the

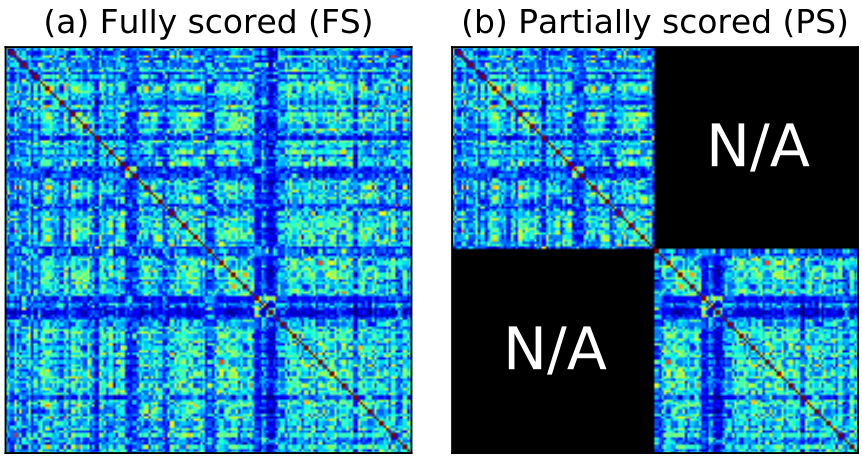

Fig. 6.9. (a) Fully scored (FS) and (b) partially scored (PS) settings

query selection using the trained speaker encoder. The number of queries per one active learning iteration was set to 43 empirically. The number of active learning iterations was 115. Other conditions were the same as the previously described ones.

6.3.2 Analysis of Perceptual Similarity Scores

The perceptual similarity scores that made the similarity matrix shown in Fig. 6.5(a) were analyzed. Figure 6.10 shows a histogram of all the scores. Approximately 70% of the scores are smaller than zero. Figure 6.11 shows a histogram of speaker-pairwise scores of the 13 unseen speakers. The score distributions of dissimilar speaker pairs (e.g., "F001-F009") have a lower variance than those of similar ones (e.g., "F010-F011"). These results suggest that the listeners easily find dissimilar speakers rather than similar ones. The similarity matrix is available online[*2].

6.3.3 Evaluation in Deep Speaker Representation Learning

This section investigated whether the proposed representation learning algorithms learn speaker representations that consider perceptual speaker similarity and achieve high-quality multi-speaker statistical speech synthesis. The following four algorithms were compared:

- **d-vec.:** Minimizing Eq. (5.4) [27]
- **Prop. (vec):** Minimizing Eq. (6.1)

[*2] `http://sython.org/demo/JSPS-DC1/index.html`

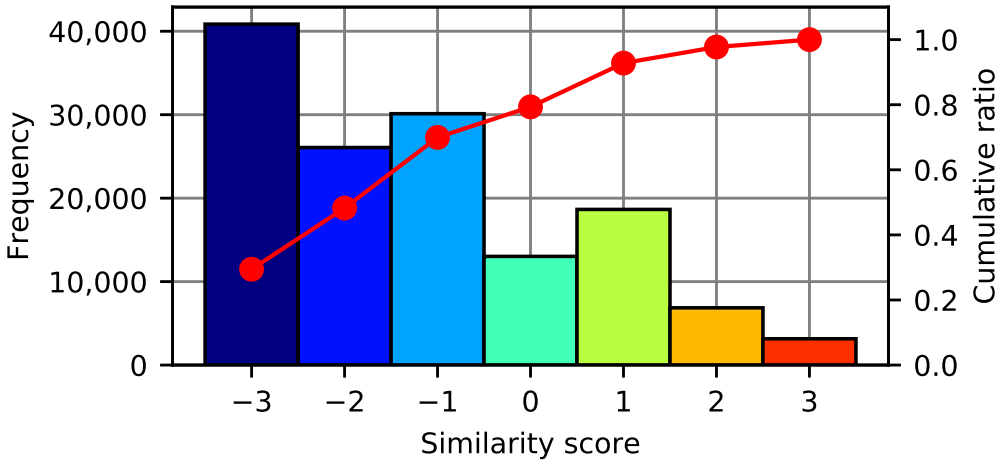

Fig. 6.10. Histogram of perceptual similarity scores of 153 female Japanese speakers. Red line denotes cumulative ratio.

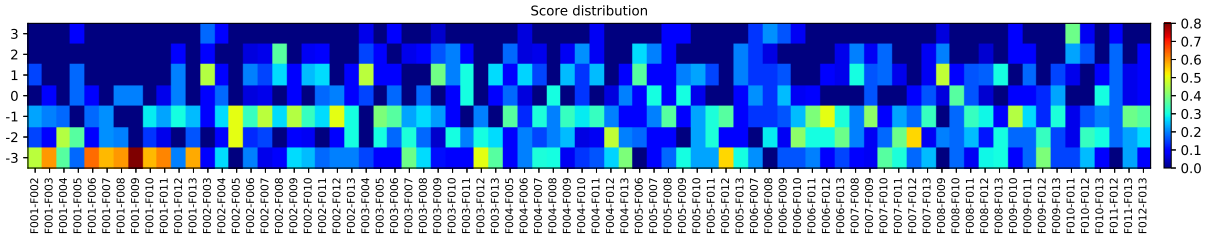

Fig. 6.11. Histogram of perceptual similarity scores of 13 speakers (from "F001" to "F013")

- **Prop. (mat):** Minimizing Eq. (6.2)
- **Prop. (graph):** Minimizing Eq. (6.6)

Correlation Analysis of Speaker Representations

The Pearson correlation coefficient between the normalized similarity scores $s_{i,j}$ and predicted similarity, i.e., values of the kernel function $k(\boldsymbol{d}_i, \boldsymbol{d}_j)$ in "d-vec.," "Prop. (vec)," and "Prop. (mat)" or the link probability $p_{i,j}$ in "Prop. (graph)," was calculated. Figure 6.12 shows the scatter plots of the similarity scores and predicted similarity with their correlation coefficients. "Prop. (*)" learn speaker representations that have a stronger correlation with the similarity scores than "d-vec.," demonstrating that the proposed speaker representations consider perceptual similarity among speakers. Among the four algorithms, "Prop. (graph)" achieves the strongest correlation not only in (a) "Seen-Seen" but also in (b) "Seen-Unseen" speaker-pair cases. This result indicates that the graph-embedding-based learning algorithm works the best to learn pairwise relations among speakers.

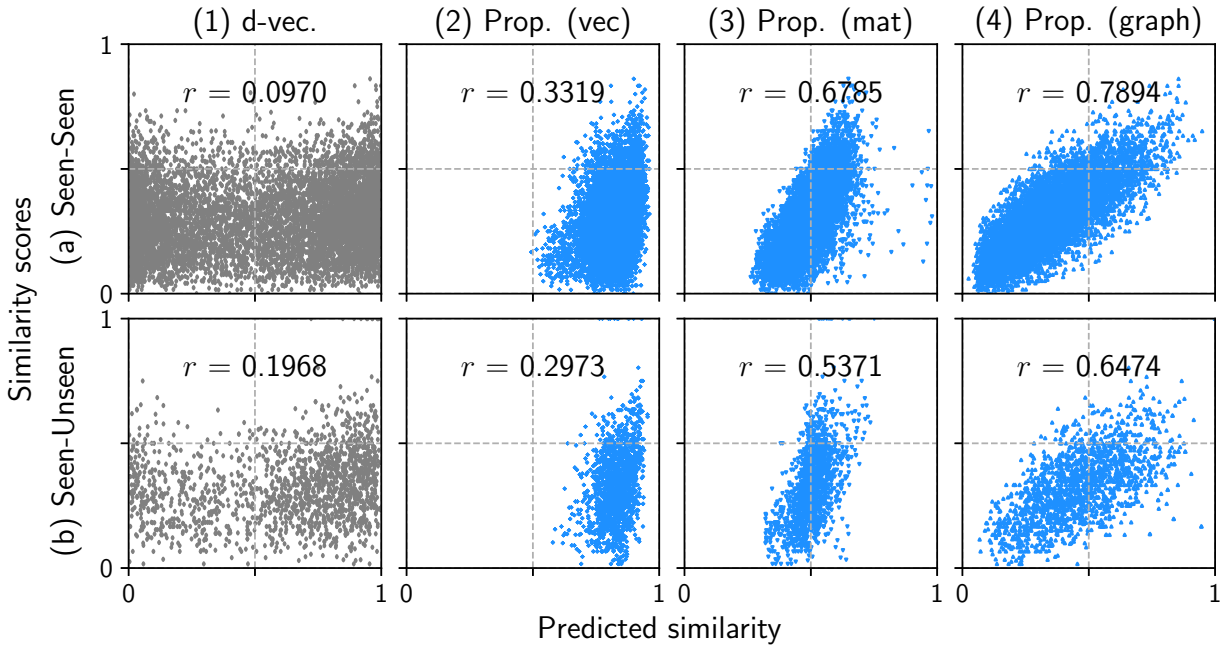

Fig. 6.12. Scatter plots of similarity scores and predicted similarity with their correlation coefficient r

Performance in Similar Speaker-pair Detection

A receiver operating characteristic (ROC) curve [134] of a binary classifier that detects similar speaker pairs using speaker representations was created. An ROC curve represents the performance of a binary classifier as a true positive rate against a false positive rate at various threshold value settings. The closer the curve follows the upper left corner (i.e., a false positive rate of zero and a true positive rate of one regardless of the threshold value settings), the more accurate the classifier is. Figure 6.13 shows ROC curves of similar speaker-pair detection using speaker representations learned by the four different algorithms. Here, "similar speaker-pair" is defined as a pair of two speakers whose perceptual similarity is greater than 0. From this figure, the proposed algorithms successfully make the ROC curves closer to the upper left corner while the conventional algorithm does not.

The area under the ROC curve (AUC) [135] that quantifies the performance of a binary classifier as a scalar value between 0.5 (random classification) and 1.0 (perfect classification) was calculated. Table 6.1 shows the AUC values calculated with the ROC curves shown in Fig. 6.13. The AUC values of "d-vec." are the lowest among the four algorithms, suggesting that the conventional speaker-classification-based learning algorithm never considers perceptual similarity among speakers. On the other hand, the three proposed algorithms increase the AUC successfully, and "Prop. (graph)" achieves the AUC

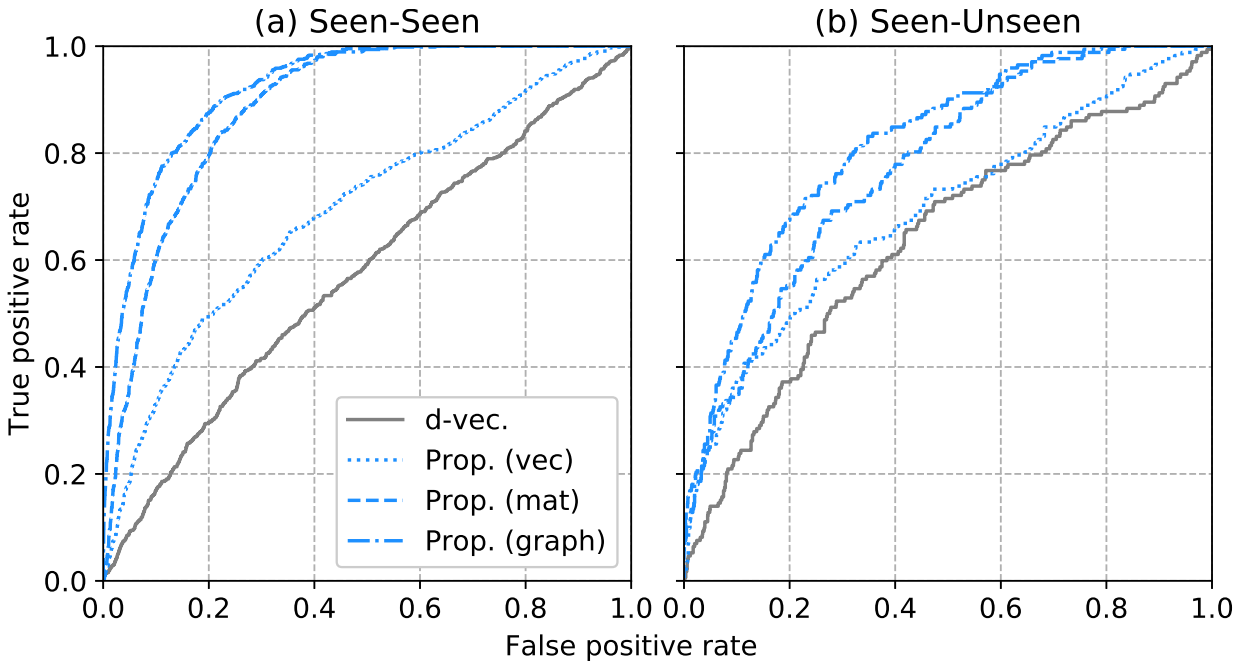

Fig. 6.13. ROC curves of similar speaker-pair detection using speaker representations. When curve becomes closer to left corner, speaker representations can be used to find similar speaker pairs more accurately.

Table 6.1. AUC values of similar speaker-pair detection using speaker representations learned by four different algorithms

	d-vec.	Prop. (vec)	Prop. (mat)	Prop. (graph)
Seen-Seen	0.5711	0.6909	0.8847	**0.9204**
Seen-Unseen	0.6347	0.6909	0.7712	**0.8157**

higher than 0.8 even in the "Seen-Unseen" speaker-pair case. These results demonstrate that the proposed algorithms construct the speaker space where similar speaker pairs are accurately found using their speaker representations.

Subjective Evaluation in VAE-based Multi-speaker Speech Synthesis

The effectiveness of the proposed speaker representations was investigated in speaker adaptation of the VAE-based multi-speaker acoustic model. The speaker adaptation aimed to reconstruct the 13 unseen speakers' voices using their speaker representations and the trained VAEs. Subjective evaluations on naturalness and speaker similarity of unseen speakers' synthetic speech were conducted. Fifty utterances of each unseen speaker were used for the speaker representation extraction. Speech samples were synthesized using mel-cepstral coefficients generated by the trained VAEs with the four different speaker encoders. Preference AB tests on the naturalness were conducted to compare the con-

Table 6.2. Preference scores on naturalness of synthetic speech (left: conventional d-vector, right: proposed algorithm). **Bold** indicates more preferred method with p-value < 0.05

Speaker	Prop. (vec)	Prop. (mat)	Prop. (graph)
F001	0.408 - **0.592**	0.456 - **0.544**	0.388 - **0.612**
F002	0.456 - **0.544**	0.456 - **0.544**	0.444 - **0.556**
F003	0.416 - **0.584**	0.444 - **0.556**	0.444 - **0.556**
F004	0.452 - **0.548**	0.460 - 0.540	0.448 - **0.552**
F005	0.380 - **0.620**	0.484 - 0.516	0.432 - **0.568**
F006	0.400 - **0.600**	0.452 - **0.548**	0.448 - **0.552**
F007	0.424 - **0.576**	0.484 - 0.516	0.424 - **0.576**
F008	0.436 - **0.564**	0.384 - **0.616**	0.392 - **0.608**
F009	0.428 - **0.572**	0.492 - 0.508	0.364 - **0.636**
F010	0.436 - **0.564**	0.464 - 0.536	0.448 - **0.552**
F011	0.460 - 0.540	0.428 - **0.572**	0.312 - **0.688**
F012	0.436 - **0.564**	0.460 - 0.540	0.440 - **0.560**
F013	0.428 - **0.572**	0.436 - **0.564**	0.372 - **0.628**

ventional algorithm ("d-vec.") with any of the three proposed algorithms ("Prop. (*)"). Twenty-five listeners participated in each of the following evaluations by using crowdsourced evaluation systems. Each listener evaluated 10 speech samples randomly extracted from the 50 utterances of each unseen speaker. Similarly, a series of preference XAB tests on the speaker similarity were conducted using the natural speech of the unseen speakers as the reference speech samples "X." The total number of task sets was 2 (AB or XAB) $\times$ 3 ("d-vec." vs. "Prop. (*)") $\times$ 13 (unseen speakers) $\times$ 25 (listeners per one task set) = 1,950.

Tables 6.2 and 6.3 list the preference scores on the naturalness and speaker similarity, respectively. "Prop. (vec)" and "Prop. (graph)" always achieve higher scores than "d-vec." regarding both the naturalness and speaker similarity. These results indicate that the proposed similarity-aware speaker representations improve synthetic speech quality in the speaker adaptation. "Prop. (mat)" also improves the naturalness; however, it significantly degrades the speaker similarity in a number of cases (e.g., "F005" and "F012"). One of the reasons may be the algorithm's trend in overfitting to seen speakers since it directly utilizes the similarity scores to determine the speakers' positions in the speaker space.

A five-point scale MOS test on the naturalness and DMOS test on the speaker similarity were conducted to compare the three proposed algorithms. Fifty listeners participated in each of the following evaluations by using crowdsourced evaluation systems. Each listener evaluated 30 speech samples randomly extracted from the 650 (50 $\times$ 13) utterances. This

Table 6.3. Preference scores on speaker similarity of synthetic speech (left: conventional d-vector, right: proposed algorithm). **Bold** indicates more preferred method with p-value < 0.05

Speaker	Prop. (vec)	Prop. (mat)	Prop. (graph)
F001	0.436 - **0.564**	0.488 - 0.512	0.412 - **0.588**
F002	0.468 - 0.532	0.496 - 0.504	0.384 - **0.616**
F003	0.432 - **0.568**	0.504 - 0.496	0.440 - **0.560**
F004	0.380 - **0.620**	0.404 - **0.596**	0.448 - **0.552**
F005	0.428 - **0.572**	**0.616** - 0.384	0.432 - **0.568**
F006	0.428 - **0.572**	0.444 - **0.556**	0.376 - **0.624**
F007	0.492 - 0.508	**0.568** - 0.432	0.452 - **0.548**
F008	0.424 - **0.576**	0.500 - 0.500	0.400 - **0.600**
F009	0.400 - **0.600**	0.500 - 0.500	0.428 - **0.572**
F010	0.432 - **0.568**	0.404 - **0.596**	0.420 - **0.580**
F011	0.348 - **0.652**	0.444 - **0.556**	0.356 - **0.644**
F012	0.492 - 0.508	**0.544** - 0.456	0.484 - 0.516
F013	0.372 - **0.628**	**0.564** - 0.436	0.436 - **0.564**

Table 6.4. Results of MOS test on naturalness and DMOS test on speaker similarity with their 95% confidence intervals (comparison among three proposed algorithms)

	Prop. (vec)	Prop. (mat)	Prop. (graph)
MOS	3.21±0.09	3.16±0.09	3.23±0.09
DMOS	2.98±0.10	2.82±0.09	2.90±0.10

evaluation enables comparing average performances of the three proposed algorithms. The total number of task sets was 2 (MOS or DMOS) $\times$ 50 (listeners per one task set) = 100. Table 6.4 shows the MOS and DMOS evaluation results. "Prop. (graph)" achieves the highest MOS among the three algorithms, although there are no significant differences among the scores. "Prop. (vec)" outperforms the others regarding the DMOS, and there is a significant difference between the scores of "Prop. (vec)" and "Prop. (mat)."

Subjective Evaluation in Speaker Interpolation

The effectiveness of the proposed speaker representations was investigated in speaker interpolation [136], a technique to artificially produce new voice characteristics by mixing two (or more) speakers' voices. Better speaker interpolation should satisfy high naturalness and high controllability of interpolated speech. In other words, it should not deteriorate the speech quality and should provide a way to control the interpolated voice characteristics intuitively. The conventional and proposed speaker representations were evaluated in speaker interpolation using a convex combination of speaker representations

Table 6.5. Results of MOS test on naturalness of interpolated speech with their 95% confidence intervals. **Bold** scores are significantly higher than those of d-vec. ($p < 0.05$)

	d-vec.	Prop. (vec)	Prop. (mat)	Prop. (graph)
F033-F134	2.88±0.13	3.07±0.13	**3.10±0.13**	**3.16±0.13**
F017-F149	3.29±0.15	3.32±0.14	3.32±0.14	3.38±0.14

to interpolate their voice characteristics [137, 138]. Formally, an interpolated representation is calculated as $\boldsymbol{d}_{\mathrm{AB}} = (1-\alpha)\boldsymbol{d}_{\mathrm{A}} + \alpha\boldsymbol{d}_{\mathrm{B}}$ using two speaker representations $\boldsymbol{d}_{\mathrm{A}}$ and $\boldsymbol{d}_{\mathrm{B}}$, where $0 \leq \alpha \leq 1$ is an interpolation coefficient. In the speaker interpolation evaluation, two speaker pairs, 1) the most *dissimilar* pairs ("F033-F134") and 2) the most *similar* ones ("F017-F149"), were considered to be mixed with $\alpha \in \{0.0, 0.25, 0.5, 0.75, 1.0\}$.

A five-point scale MOS test on naturalness of interpolated speech ($\alpha = 0.5$) was conducted to compare speaker representations learned by the four different algorithms. Fifty listeners participated in the MOS test by using crowdsourced evaluation systems. Each listener evaluated 16 samples of the interpolated speech. The total number of task sets was 2 ("F033-F134" or "F017-F149") × 50 (listeners per one task) = 100. Table 6.5 lists results of the MOS test. From the results of the interpolation using the most dissimilar speaker pair ("F033-F134"), the three proposed algorithms achieve higher MOS values than the conventional d-vector, and the similarity matrix and graph embedding significantly outperform the d-vector. "Prop. (*)" also outperform "d-vec." in the interpolation using the most similar speaker pair ("F017-F149"), although there are no significant differences among the MOS values. These results indicate that the proposed speaker representations improve the synthetic speech quality not only in speaker adaptation but also in speaker interpolation.

A variant of the preference XAB test was conducted to evaluate the speaker similarity of interpolated speech. In the evaluation, listeners first played three speech samples interpolated with different coefficients: "X" ($\alpha \in \{0.0, 0.25, 0.5, 0.75, 1.0\}$), "A" ($\alpha = 0.0$), and "B" ($\alpha = 1.0$), and then answered which of the two samples "A" or "B" sounded similar to "X." Thirty listeners participated in the evaluation by using crowdsourced evaluation systems. Each listener evaluated 20 speech samples. The total number of task sets was 2 ("F033-F134" or "F017-F149") × 30 (listeners per one task) = 60. Figure 6.14 shows the preference score curves against the five different interpolation coefficients. The shapes of the curves significantly change between Figs. 6.14(a) and (b). This result suggests that perceptual similarity of a speaker pair greatly affects the result of speaker interpolation; i.e., the more dissimilar a speaker pair is, the larger the difference of the interpolated

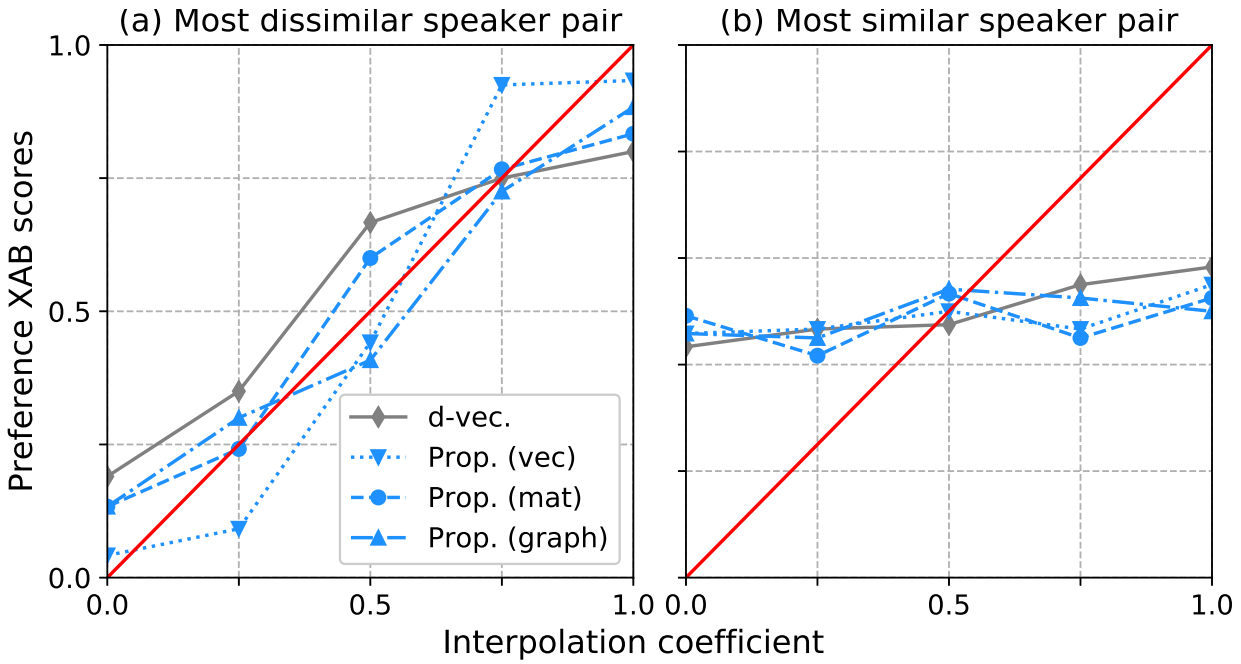

Fig. 6.14. Results of XAB tests on speaker similarity of the interpolated speech using (a) most dissimilar speaker pair ("F033-F134") and (b) most similar one ("F017-F149"). When score curves become closer to red lines, listeners accurately infer two speakers' mixing ratio from interpolated speech.

speech becomes. Red lines representing "interpolation coefficients and preference XAB scores are equal" were added to Fig. 6.14 for illustrating this observation clearly. When the score curves become closer to the red lines, listeners accurately infer two speakers' mixing ratio from interpolated speech. All the preference scores shown in Fig. 6.14(b) are near 0.5 regardless of the interpolation coefficient settings, i.e., listeners hardly perceived the differences among the interpolated speech samples. This is a natural result because very similar speakers' voices are mixed, and listeners therefore cannot detect the difference between the two speech samples "A" ($\alpha = 0.0$) and "B" ($\alpha = 1.0$). One can quantify the controllability of speaker interpolation using a pair of dissimilar speakers as the distance between the red line and the score curve in Fig. 6.14(a). The linear least squares error between the red line and preference scores were calculated. The calculation results of "d-vec.," "Prop. (vec)," "Prop. (mat)," and "Prop. (graph)" were 0.1139, 0.0654, 0.0559, and 0.0429, respectively. Therefore, the consideration of perceptual similarity among speakers in deep speaker representation learning improves not only the quality but also the controllability of synthetic speech in the VAE-based multi-speaker speech synthesis.

6.3.4 Evaluation of Active Learning

The active learning's effectiveness was investigated using each of the three proposed representation learning algorithms independently. In addition to the three query strategies described in Section 6.2.3 (i.e., "LSF", "HSF," and "USF"), "FS" and "PS" were compared. The former and latter trained a speaker encoder using the fully observed scores and partially observed ones shown in Figs. 6.9(a) and (b), respectively.

Evaluation of AUC Improvement

How the proposed active learning affected the AUC of similar speaker-pair detection was investigated. Figure 6.15 shows the AUC curves against the active learning iterations. Red and blue lines denote the final AUC values of "FS" and "PS," respectively. Note that the final AUC values of "LSF," "HSF," and "USF," in Fig. 6.15(a) does not necessarily correspond to those of "FS" because their speaker encoders sequentially learn perceptual similarity among the 140 seen speakers using differently ordered similarity scores. The query strategies significantly affect the AUC improvement by the active learning, and "USF" reasonably works among the three strategies regardless of the training algorithm differences. Active learning using "Prop. (vec)" or "Prop. (graph)" successfully improves the AUC through the iterations better than "PS" in both the "Seen-Seen" and "Seen-Unseen" speaker-pair cases. Meanwhile, "Prop. (mat)" clearly shows its trend in overfitting to the seen speaker pairs; i.e., it increased the AUC by the active learning iterations in the "Seen-Seen" speaker-pair case but results in decreasing the AUC in the "Seen-Unseen" case.

Evaluation of Synthetic Speech Quality

Whether the active learning efficiently trained a speaker encoder that improved synthetic speech quality while reducing the scoring and training costs was investigated. Similar to Section 6.3.3, five-point scale MOS and DMOS tests were conducted to compare the quality of synthetic speech made by speaker representations of "FS," "PS," and "USF." The active learning using "USF" was performed with three different iterations to increase the percentage of additionally scored speaker pairs: 25% (30 iterations), 50% (60 iterations), and 75% (90 iterations). Tables 6.6 and 6.7 list the results of the MOS and DMOS tests, respectively. "USF" achieves synthetic speech quality comparable to that of "FS" with the fewer number of additional similarity score observations and active learning iterations. This result demonstrates that active learning of the perceptual-similarity-aware

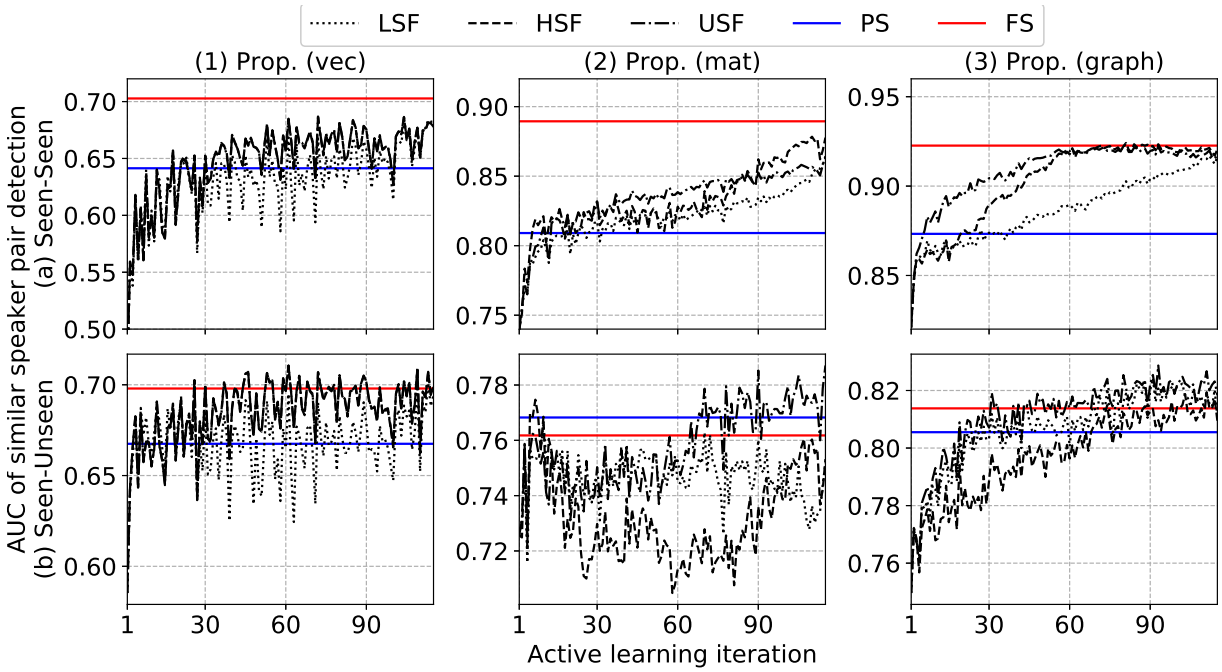

Fig. 6.15. Curves of similar speaker-pair detection AUC against active learning iteration

Table 6.6. Results of MOS test on naturalness of synthetic speech with their 95% confidence intervals (proposed algorithms using active learning). Second column denotes percentages of number of additionally scored speaker pairs compared with "PS." **Bold** scores are comparable to those of "FS" ($p > 0.05$)

		Prop. (vec)	Prop. (mat)	Prop. (graph)
PS	(0%)	2.91±0.14	**2.98±0.13**	2.94±0.14
USF	(25%)	2.99±0.12	**2.97±0.13**	**3.11±0.13**
	(50%)	**3.11±0.13**	**3.02±0.13**	**3.12±0.14**
	(75%)	**3.18±0.13**	**3.04±0.13**	**3.13±0.14**
FS	(100%)	3.19±0.13	3.02±0.12	3.18±0.13

speaker representations effectively reduces the number of scoring times while achieving higher synthetic speech quality with fewer training iterations. Focusing on the results of "Prop. (mat)," there are no significant differences among the five scores, and "FS" marks the lowest scores among the three proposed algorithms. One of the causes may be the overfitting trend observed in Fig. 6.15(b)(2).

6.3.5 Visualization of Speaker Space

Speaker spaces constructed by speaker representations of the four algorithms were visualized. The purpose of this visualization is to investigate how the four different algorithms affect the speaker space. Figure 6.16 shows the PCA plots of the speaker representations.

Table 6.7. Results of DMOS test on speaker similarity of synthetic speech with their 95% confidence intervals (proposed algorithms using active learning). Second column denotes percentages of number of additionally scored speaker pairs compared with "PS." **Bold** scores are comparable to "FS" ($p > 0.05$)

		Prop. (vec)	Prop. (mat)	Prop. (graph)
PS	(0%)	2.85±0.14	**2.90±0.13**	2.86±0.13
USF	(25%)	**2.95±0.14**	**2.93±0.13**	**3.03±0.13**
	(50%)	**3.04±0.14**	**3.00±0.13**	**3.02±0.13**
	(75%)	**3.05±0.14**	**3.03±0.13**	**3.06±0.13**
FS	(100%)	3.14±0.14	2.98±0.13	3.08±0.14

From this figure, the speaker space of "d-vec." is the widest among the four plots, and it completely ignores the perceptual similarity among the speakers. On the other hand, the three proposed algorithms construct the speaker spaces where the speakers are densely distributed and the perceptual similarity is preserved. However, "Prop. (mat)" tends to excessively separate disjoint speaker clusters, i.e., "F055–F058," from the other, which may be one possible cause of the poor generalization shown in Fig. 6.15 and speech quality degradation.

6.4 Summary

This chapter proposed novel algorithms for incorporating perceptual similarity among speakers into deep speaker representation learning. The proposed speaker representation learning algorithms utilize a perceptual speaker similarity matrix obtained by large-scale perceptual scoring as the target for the speaker encoder training. The algorithms learn speaker representations with three different usages of the matrix: a set of vectors, the Gram matrix, and a graph. This chapter further proposed an active learning algorithm to reduce costs of scoring and training. The active learning algorithm iterates the perceptual similarity scoring and speaker encoder training. Queries in this algorithm are generated from the sequentially-trained speaker encoder for prioritizing unscored speaker-pairs to be scored next. The experimental evaluation results demonstrated that 1) the proposed speaker representation learning algorithms learned speaker representations strongly correlated with perceptual similarity scores, 2) the representations improved synthetic speech quality better than conventional speaker-classification-derived d-vectors, and 3) the proposed active learning algorithm achieved higher synthetic speech quality while reducing costs of scoring and training.

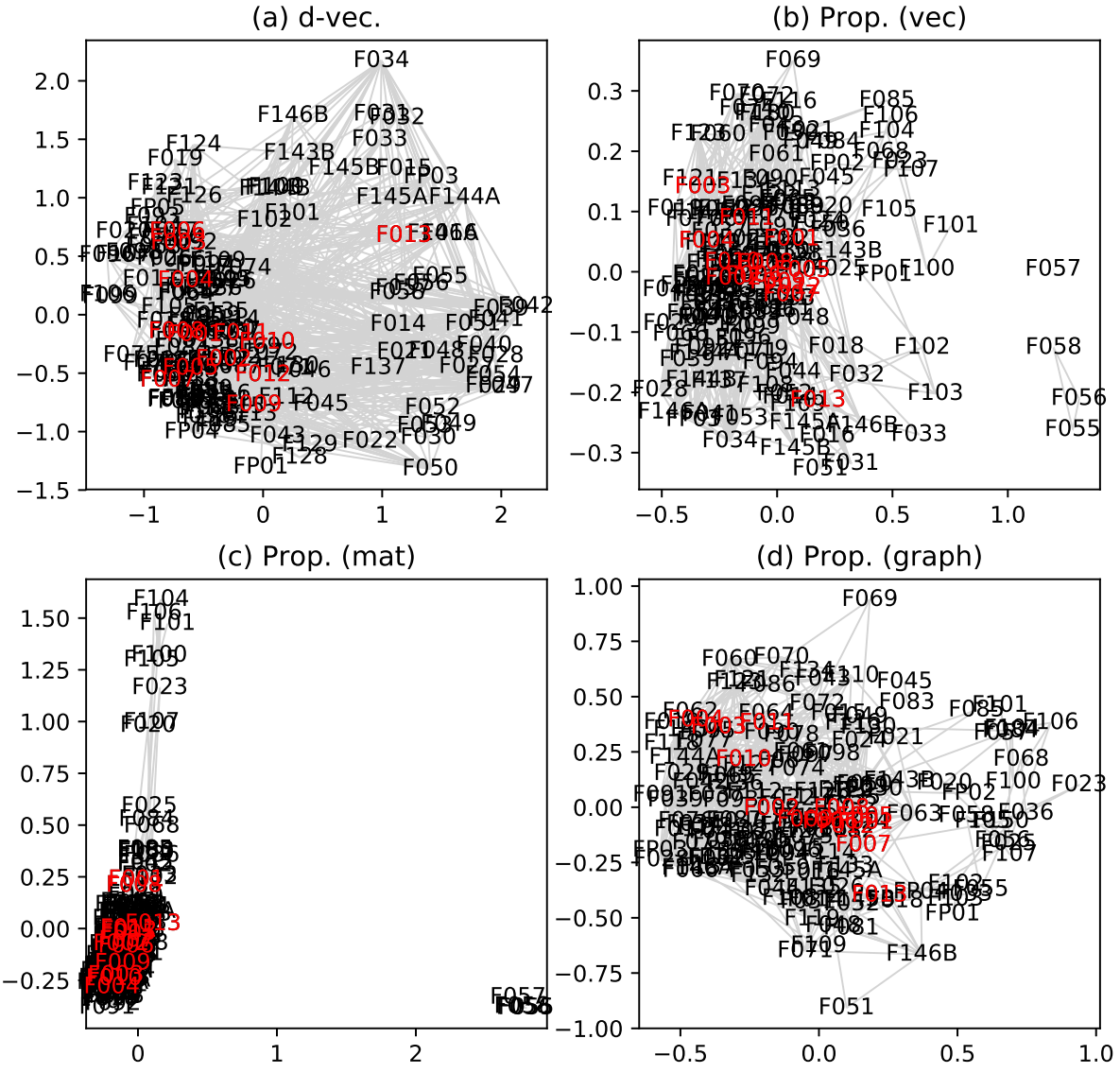

Fig. 6.16. PCA plots of speaker representations learned by (a) d-vec., (b) Prop. (vec), (c) Prop. (mat), and (d) Prop. (graph). Similar-speaker pairs are connected by gray edges. Red points denote unseen speakers. PCA was applied to 153 speakers' embedding learned by each method independently.

Chapter 7

Conclusion

7.1 Summary of This Thesis

This thesis addressed the issues of conventional DNN-based speech synthesis for synthesizing high-quality, versatile, and intuitively controllable speech. The following three issues were solved: 1) low-quality synthetic speech due to over-smoothing, 2) limited speaker diversity in synthetic speech, and 3) uninterpretable speaker representation.

Chapter 2 briefly reviewed the basic framework of DNN-based speech synthesis. The framework's four crucial factors were described: 1) feature analysis, 2) acoustic modeling, 3) speech parameter generation, and 4) speech waveform synthesis.

Chapter 3 presented the proposed GAN-based method to improve synthetic speech quality. The basic framework of GANs was first described. The introduction of GANs to train DNNs for speech synthesis was then discussed. From the divergence minimization perspective, the effects of the divergence in improving synthetic speech quality were investigated. Experimental evaluations were conducted to demonstrate this method's effectiveness in TTS and VC using vocoder parameters. The results indicated that 1) this method generated natural speech parameters regardless of its hyperparameter settings, and 2) W-GAN minimizing the earth-mover's distance worked the best among several GANs in terms of improving the synthetic speech quality.

Chapter 4 extended the proposed GAN-based method described in Chapter 3 to DNN-based speech synthesis using STFT spectra. A simple but effective approach was presented to overcome the difficulty in modeling high-dimensional and complicated amplitude spectra based on GANs. Various frequency scales that are related to human speech perception were introduced to this GAN-based method. The effectiveness of this method was evaluated in TTS using STFT spectra. The results indicated that 1) GANs using low-frequency-

resolution amplitude spectra improved speech quality and worked robustly against the settings of the frequency resolution and hyperparameters, 2) comparing low-, original-, and multi-frequency-resolution amplitude spectra, the use of low-frequency-resolution ones worked best to improve synthetic speech quality, and 3) the use of the inverse mel frequency scale for obtaining low-frequency-resolution amplitude spectra further improved synthetic speech quality.

Chapter 5 presented the proposed high-quality and versatile speech synthesis method based on VAEs. The VAE-based speech parameter generation process was mathematically formulated to explicitly model the phonetic content and speaker individuality as latent variables. Non-parallel and many-to-many VC that can reproduce and transform arbitrary speaker's voice characteristics was established using VAEs. The trade-off between the number of seen speakers and dimensionality of continuous speaker representation with this method was investigated. The effectiveness of this method was objectively and subjectively evaluated in VC. The results indicated that 1) the introduction of the DNN-based speech recognition model contributed to significant quality improvement in converted speech, 2) the use of continuous speaker representations achieved high-quality VC even if the source and target speakers are unseen during the VAE training, and 3) high-dimensional speaker representation did not necessarily improve the converted speech quality but a large number of seen speakers consistently did improve this.

Chapter 6 presented the proposed perceptual-similarity-aware speaker representation learning method for increasing the interpretability of speaker representations. The subjective scoring of perceptual speaker-pair similarity to obtain a perceptual speaker similarity matrix was first described. Deep speaker representation learning algorithms using a loss function defined by the similarity matrix were then presented. An active learning algorithm to reduce the costs of scoring and training was finally introduced. This method's effectiveness was evaluated with the proposed VAE-based multi-speaker speech synthesis method described in Chapter 5. The results indicated that 1) the proposed speaker representation learning method learned speaker representations strongly correlated with perceptual similarity scores, 2) the representations improved synthetic speech quality better than the conventional representations derived from a DNN-based speaker recognition model, and 3) the active learning algorithm achieved higher synthetic speech quality while reducing the costs of scoring and training.

7.2 Future Directions

7.2.1 Improving Synthetic Speech Quality Further

The proposed methods successfully improved synthetic speech quality compared with the conventional methods. However, several subjective evaluation results (e.g., Fig. 3.21 and Table 5.1) revealed that the MOS values on naturalness of synthetic speech were around 3.0, which indicates that there is a still gap between natural speech (MOS $>$ 4.0) and synthetic speech. Therefore, I will aim to reduce the gap and to achieve communication using the speech synthesis technology for synthesizing speech almost indistinguishable from natural speech. One solution is to use recently developed neural vocoders (e.g., WaveNet vocoder [139]), which can synthesize high-fidelity speech waveforms from speech parameters using DNNs.

7.2.2 Implementing with Real-time Speech Synthesis

Daily speech communication in most cases takes place in real time. However, the proposed methods do not consider such real-time situations, which limits their application ranges. Therefore, I will introduce a DNN-based real-time speech synthesis method, such as incremental TTS [140] and real-time VC [141, 142], into the proposed methods and evaluate their effectiveness.

7.2.3 Extending to Multilingual Speech Synthesis

All experiments for this thesis were conducted using only Japanese speech corpora. To enable speech communication beyond language barriers, I will evaluate the language dependency of the proposed methods by conducting experiments using other languages, such as English and Chinese. I will also extend the proposed methods to multilingual speech synthesis [143] that can synthesize voices in multiple languages using a single acoustic model.

7.2.4 Modeling Broader Speech Perception

In Chapter 6, inter-speaker perceptual similarity was modeled by using multi-speaker corpus including reading-style speech to learn speaker representations that strongly correlate with human's perception and enable intuitively controllable speech synthesis. I will extend

this idea to model human's broader speech perception, such as various speaking styles, emotions, and different languages. This will enable developing more controllable speech synthesis systems that can be easily tuned through human-computer interaction.

Publications and Research Activities

Publications Related to This Thesis

Original Journal Papers

1. **Yuki Saito**, Shinnosuke Takamichi, and Hiroshi Saruwatari, "Perceptual-similarity-aware deep speaker representation learning for multi-speaker generative modeling," *IEEE/ACM Transactions on Audio, Speech, and Language Processing,* vol. xx, no. x, pp. xx-xx, 2021. (ACCEPTED, corresponds to Chapter 6)
2. **Yuki Saito**, Taiki Nakamura, Yusuke Ijima, Kyosuke Nishida, and Shinnosuke Takamichi, "Non-parallel and many-to-many voice conversion using variational autoencoders integrating speech recognition and speaker verification," *Acoustical Science and Technology,* vol. 42, no. 1, pp. 1–11, Jan. 2021. (corresponds to Chapter 5 and Appendix Appendix C)
3. **Yuki Saito**, Kei Akuzawa, and Kentaro Tachibana, "Joint adversarial training of speech recognition and synthesis models for many-to-one voice conversion using phonetic posteriorgrams," *IEICE Transactions on Information and Systems,* vol. 103-D, no. 9, pp. 1978–1987, Sep. 2020. (corresponds to Chapter 3 and Appendix Appendix B)
4. **Yuki Saito**, Shinnosuke Takamichi, and Hiroshi Saruwatari, "Vocoder-free text-to-speech synthesis incorporating generative adversarial networks using low-/multi-frequency STFT amplitude spectra," *Computer Speech and Language,* vol. 58, pp. 347–363, Nov. 2019. (corresponds to Chapter 4)
5. **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. (corresponds to Chapter 3)

6. **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. (corresponds to Chapter 3 and Appendix Appendix A)

International Conferences (Peer-Reviewed)

1. **Yuki Saito**, Shinnosuke Takamichi, and Hiroshi Saruwatari, "DNN-based speaker embedding using subjective inter-speaker similarity for multi-speaker modeling in speech synthesis," *Proc. SSW,* pp. 51–56, Vienna, Austria, Sep. 2019. (corresponds to Chapter 6)
2. **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,* pp. 5274–5278, Alberta, Canada, Apr. 2018. (corresponds to Chapter 5)
3. **Yuki Saito**, Shinnosuke Takamichi, and Hiroshi Saruwatari, "Text-to-speech synthesis using STFT spectra based on low-/multi-resolution generative adversarial networks," *Proc. ICASSP,* pp. 5299–5303, Alberta, Canada, Apr. 2018. (corresponds to Chapter 4)
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. (corresponds to Chapter 3)

Technical Reports

1. **Yuki Saito**, Shinnosuke Takamichi, and Hiroshi Saruwatari, "Active learning for DNN-based speaker embedding considering subjective inter-speaker similarity," *IPSJ SIG Technical Report,* 2021-SLP-xxx, no. x, pp. xx–xx, Mar. 2021. (to appear in Japanese, corresponds to Chapter 6)
2. **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,* SP2017-88, Vol. 117, No. 517, pp. 21–26, Mar. 2018. (in Japanese, corresponds to Chapter 5)
3. **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, corresponds to Chapter 3)

4. **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, corresponds to Chapter 3 and Appendix Appendix A)

Domestic Conferences

1. **Yuki Saito**, Shinnosuke Takamichi, and Hiroshi Saruwatari, "DNN-based speaker embedding using graph embedding of subjective inter-speaker similarity," *Proc. ASJ, Autumn meeting,* 1-2-4, pp. 697–698, Sep. 2020. (in Japanese, corresponds to Chapter 6)
2. **Yuki Saito**, Kei Akuzawa, and Kentaro Tachibana, "Joint adversarial training algorithm of speech recognition and synthesis models for many-to-one voice conversion using phonetic posteriorgrams," *Proc. ASJ, Autumn meeting,* 2-4-2, pp. 963–966, Sep. 2019. (in Japanese, corresponds to Chapter 3 and Appendix Appendix B)
3. **Yuki Saito**, Shinnosuke Takamichi, and Hiroshi Saruwatari, "Evaluation of DNN-based multi-speaker speech synthesis using DNN-based speaker embedding considering subjective inter-speaker similarity," *Proc. ASJ, Autumn meeting,* 1-P-18, pp. 999–1002, Sep. 2019. (in Japanese, corresponds to Chapter 6)
4. **Yuki Saito**, Shinnosuke Takamichi, and Hiroshi Saruwatari, "DNN-based speaker embedding considering subjective inter-speaker similarity towards DNN-based speech synthesis," *Proc. ASJ, Spring meeting,* 3-10-7, pp. 1067–1070, Mar. 2019. (in Japanese, corresponds to Chapter 6)
5. Taiki Nakamura, **Yuki Saito**, Kyosuke Nishida, Yusuke Ijima, and Shinnosuke Takamichi, "Evaluation of VAE-based non-parallel and many-to-many voice conversion conditioned by phonetic posteriorgrams and d-vectors in terms of training data and dimensionality of d-vectors," *Proc. ASJ, Spring meeting,* 2-P-30, pp. 1149–1150, Mar. 2019. (in Japanese, corresponds to Chapter 5)
6. **Yuki Saito**, Shinnosuke Takamichi, and Hiroshi Saruwatari, "Adversarial DNN-based speech synthesis using multi-frequency-resolution STFT spectra," *Proc. ASJ, Spring meeting,* 3-8-14, pp. 259–262, Mar. 2018. (in Japanese, corresponds to Chapter 4)
7. **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, corresponds to Chapter 3)

8. **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, corresponds to Chapter 3 and Appendix Appendix A)
9. **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, corresponds to Chapter 3)
10. **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, corresponds to Chapter 3)

Awards

1. 2020 IEEE Signal Processing Society Young Author Best Paper Award, Jun. 2021.
2. The 49th Awaya Prize Young Researcher Award of ASJ, Mar. 2021.
3. Outstanding Paper Award for Young C&C Researchers, Jan. 2019.
4. The 12th IEEE Signal Processing Society Japan Student Journal Paper Award, Nov. 2018.
5. 2017 IEICE ISS Young Researcher's Award in Speech Field, Aug. 2018.
6. Partial Exemption from Repayment of Scholarship Loan for Students with Outstanding Results, Japan Student Services Organization (JASSO), May 2018.
7. The 33rd TELECOM System Technology Award for Students from TAF, Mar. 2018.
8. The 1st IEEE Signal Processing Society Tokyo Joint Chapter Student Award, Nov. 2017.
9. Spoken Language Processing Student Grant Award of ICASSP, Mar. 2017.
10. The 14th Best Student Presentation Award of Acoustical Society of Japan, Mar. 2017.
11. 2017 IEICE ISS Student Poster Award, Jan. 2017.

Competitive Funds

1. Grant-in-Aid for JSPS Fellows, the Japan Society of the Promotion of Science (JSPS), May 2018.

2. Grants for Researchers Attending International Conferences from NEC C&C, Apr. 2018.

Misc.

1. I was invited to Google Speech Technology Summit 2018, Google London, United Kingdom. I talked about two research papers that were accepted to ICASSP 2018 at the poster session.
2. A figure taken from our paper appeared on the cover of the IEEE/ACM Transactions on Audio, Speech, and Language Processing (January/February issue).

Other Publications

Original Journal Papers

1. Takaaki Saeki, **Yuki Saito**, Shinnosuke Takamichi, and Hiroshi Saruwatari, "Real-time full-band voice conversion with sub-band modeling and data-driven phase estimation of spectral differentials," *IEICE Transactions on Information and Systems,* 2021. (under review, conditionally accepted)
2. Shinnosuke Takamichi, Ryosuke Sonobe, Kentaro Mitsui, **Yuki Saito**, Tomoki Koriyama, Naoko Tanji and Hiroshi Saruwatari, "JSUT and JVS: free Japanese voice corpora for accelerating speech synthesis research," *Acoustical Science and Technology,* vol. 41, No. 5, pp. 761–768, Sep. 2020.
3. Hiroki Tamaru, **Yuki Saito**, Shinnosuke Takamichi, Tomoki Koriyama, and Hiroshi Saruwatari, "Generative moment matching network-based neural double-tracking for synthesized and natural singing voices," *IEICE Transactions on Information and Systems,* vol. E103-D, no. 3, pp. 639–647, Mar. 2020.
4. Shinnosuke Takamichi, **Yuki Saito**, Norihiro Takamune, Daichi Kitamura, and Hiroshi Saruwatari, "Phase reconstruction from amplitude spectrograms based on directional-statistics deep neural networks," *Signal Processing,* vol. 169, pp. 107368, Apr. 2020.

International Conferences (Peer-Reviewed)

1. Yota Ueda, Kazuki Fujii, **Yuki Saito**, Shinnosuke Takamichi, Yukino Baba, and Hiroshi Saruwatari, "HumanACGAN: conditional generative adversarial network with human-based auxiliary classifier and its evaluation in phoneme perception,"

Proc. ICASSP, pp. xxxx–xxxx, Toronto, Canada, Jun. 2021. (ACCEPTED)

2. Yuki Yamashita, Tomoki Koriyama, **Yuki Saito**, Shinnosuke Takamichi, Yusuke Ijima, Ryo Masumura, and Hiroshi Saruwatari, "Investigating effective additional contextual factors in DNN-based spontaneous speech synthesis," *Proc. INTERSPEECH,* pp. 3201–3205, Shanghai, China, Oct. 2020.
3. Shunsuke Goto, Kotaro Ohnishi **Yuki Saito**, Kentaro Tachibana, and Koichiro Mori, "Face2Speech: towards multi-speaker text-to-speech synthesis using an embedding vector predicted from a face image," *Proc. INTERSPEECH,* pp. 2947–2951, Shanghai, China, Oct. 2020.
4. Detai Xin, **Yuki Saito**, Shinnosuke Takamichi, Tomoki Koriyama, and Hiroshi Saruwatari, "Cross-lingual text-to-speech synthesis via domain adaptation and perceptual similarity regression in speaker space," *Proc. INTERSPEECH,* pp. 1321–1325, Shanghai, China, Oct. 2020.
5. Takaaki Saeki, **Yuki Saito**, Shinnosuke Takamichi, and Hiroshi Saruwatari, "Real-time, full-band, online DNN-based voice conversion system using a single CPU," *Proc. INTERSPEECH,* pp. 1021–1022, Shanghai, China, Oct. 2020.
6. **Yuki Saito**, Shinnosuke Takamichi, and Hiroshi Saruwatari, "SMASH corpus: a spontaneous speech corpus recording third-person audio commentaries on gameplay," *Proc. LREC,* pp. 6573–6579, Marseille, France, May 2020.
7. Yuki Yamashita, Tomoki Koriyama, **Yuki Saito**, Shinnosuke Takamichi, Yusuke Ijima, Ryo Masumura, and Hiroshi Saruwatari, "DNN-based speech synthesis using abundant tags of spontaneous speech corpus," *Proc. LREC,* pp. 6440–6445, Marseille, France, May 2020.
8. Kazuki Fujii, **Yuki Saito**, Shinnosuke Takamichi, Yukino Baba, and Hiroshi Saruwatari, "HumanGAN: generative adversarial network with human-based discriminator and its evaluation in speech perception modeling," *Proc. ICASSP,* pp. 6239–6243, Barcelona, Spain, May 2020.
9. Takaaki Saeki, **Yuki Saito**, Shinnosuke Takamichi, and Hiroshi Saruwatari, "Lifter training and sub-band modeling for computationally efficient and high-quality voice conversion using spectral differentials," *Proc. ICASSP,* pp. 7784–7788, Barcelona, Spain, May 2020.
10. Taiki Nakamura, **Yuki Saito**, Shinnosuke Takamichi, Yusuke Ijima, and Hiroshi Saruwatari, "V2S attack: building DNN-based voice conversion from automatic speaker verification," *Proc. SSW,* pp. 161–165, Vienna, Austria, Sep. 2019.
11. Hiroki Tamaru, **Yuki Saito**, Shinnosuke Takamichi, Tomoki Koriyama, and Hi-

roshi Saruwatari, "Generative moment matching network-based random modulation post-filter for DNN-based singing voice synthesis and neural double-tracking," *Proc. ICASSP,* pp. 7070–7074, Brighton, U.K., May 2019.

12. 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. APSIPA-ASC,* pp. 99–103, Hawaii, U.S.A., Nov. 2018.
13. Shinnosuke Takamichi, **Yuki Saito**, Norihiro Takamune, Daichi Kitamura, and Hiroshi Saruwatari, "Phase reconstruction from amplitude spectrograms based on directional-statistics deep neural networks," *Proc. IWAENC,* pp. 286–290, Tokyo, Japan, Sep. 2018.
14. 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.

Technical Reports

1. Masaki Kurata, Shinnosuke Takamichi, Takaaki Saeki, Riku Arakawa, **Yuki Saito**, Keita Higuchi, and Hiroshi Saruwatari, "A method for obtaining speaking characteristics based on real-time DNN-based voice conversion feedback," *IPSJ SIG Technical Report,* 2021-SLP-xxx, No. x, pp. xx–xx, Mar. 2021. (to appear in Japanese)
2. Kazuki Fujii, **Yuki Saito**, Shinnosuke Takamichi, Yukino Baba, and Hiroshi Saruwatari, "HumanGAN: generative adversarial networks based on human perception evaluation and its application in speech naturalness modeling," *IEICE Technical Report,* SP2020-06, Vol. 120, No. 57, pp. 15–20, Jun. 2020. (in Japanese)
3. Satoshi Naitou, **Yuki Saito**, Shinnosuke Takamichi, Yasuyuki Saito, and Hiroshi Saruwatari, "Automatic estimation of breath position for singing VOCALOID song," *IPSJ SIG Technical Report,* 2020-MUS-127, No. 33, pp. 1–6, Jun. 2020. (in Japanese)
4. Yuki Yamashita, Tomoki Koriyama, **Yuki Saito**, Shinnosuke Takamichi, Yusuke Ijima, Ryo Masumura, and Hiroshi Saruwatari, "The effectiveness of additional context in DNN-based spontaneous speech synthesis," *IEICE Technical Report,* SP2019-61, Vol. 119, No. 441, pp. 65–70, Mar. 2020. (in Japanese)
5. Takaaki Saeki, **Yuki Saito**, Shinnosuke Takamichi, and Hiroshi Saruwatari, "Lifter

training and sub-band modeling for DNN-based voice conversion using spectral differentials," *IPSJ SIG Technical Report,* 2020-SLP-131, No. 2, pp. 1–6, Feb. 2020. (in Japanese)

6. Kazuki Fujii, **Yuki Saito**, Shinnosuke Takamichi, Yukino Baba, and Hiroshi Saruwatari, "HumanGAN: generative adversarial networks trained with human perception evaluation," *IBIS Workshop 2019,* Nov. 2019. (in Japanese)
7. Taiki Nakamura, **Yuki Saito**, Shinnosuke Takamichi, Yusuke Ijima, and Hiroshi Saruwatari, "Speaker V2S attack: statistical voice conversion built from speaker verification and its evaluation on speaker spoofing attack," *CSS 2019,* pp. 697–703, Oct. 2019. (in Japanese)
8. Shinnosuke Takamichi, Kentaro Mitsui, **Yuki Saito**, Tomoki Koriyama, Naoko Tanji, and Hiroshi Saruwatari, "JVS corpus: online available Japanese versatile speech corpus," *IPSJ SIG Technical Report,* 2019-SLP-129, No. 1, pp. 1–6, Oct. 2019. (in Japanese)
9. Hiroki Tamaru, **Yuki Saito**, Shinnosuke Takamichi, Tomoki Koriyama, and Hiroshi Saruwatari, "Generative moment matching network-based random modulation post-filter for singing voices synthesized using DNNs and its application to neural double-tracking," *IPSJ SIG Technical Report,* 2018-SLP-125, No. 1, pp. 1–6, Dec. 2018. (in Japanese)
10. Satoshi Mizoguchi, **Yuki Saito**, Shinnosuke Takamichi, and Hiroshi Saruwatari, "Evaluation of DNN-based low-musical-noise speech enhancement using kurtosis matching," *IEICE Technical Report,* EA2018-66, Vol. 118, No. 312, pp. 19–24,, Nov. 2018. (in Japanese)
11. Shinnosuke Takamichi, **Yuki Saito**, Norihiro Takamune, Daichi Kitamura, and Hiroshi Saruwatari, "Phase reconstruction from amplitude spectra based on von Mises distribution DNN," *IPSJ SIG Technical Report,* 2018-SLP-122, No. 1, pp. 1–6, Jun. 2018. (in Japanese)
12. 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)
13. 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)

Domestic Conferences

1. Yota Ueda, Kazuki Fujii, **Yuki Saito**, Shinnosuke Takamichi, Yukino Baba, and Hiroshi Saruwatari, "HumanACGAN: conditional generative adversarial network using human-based auxiliary classifier and its evaluation in representing conditional distribution of phoneme perception," *Proc. ASJ, Spring meeting,* 1-2-14, pp. xxx–xxx, Mar. 2021. (to appear in Japanese)
2. Takaaki Saeki, **Yuki Saito**, Shinnosuke Takamichi, and Hiroshi Saruwatari, "Implementation and evaluation of real-time full-band DNN-based Voice Conversion based on sub-band filtering," *Proc. ASJ, Autumn meeting,* 1-2-11, pp. 715–718, Sep. 2020. (in Japanese)
3. Kazuki Fujii, **Yuki Saito**, Shinnosuke Takamichi, Yukino Baba, and Hiroshi Saruwatari, "HumanGAN: generative adversarial network with human-based discriminator and its evaluation in naturalness modeling of speech," *Proc. ASJ, Spring meeting,* 3-P-40, pp. 1181–1184, Mar. 2020. (in Japanese)
4. Shinnosuke Takamichi, Kai Onuma, Taku Kaneda, Takashi Kaneda, **Yuki Saito**, Tomoki Koriyama, and Hiroshi Saruwatari, "Crowdsourcing-based parameter optimization for frequency warping-based speaker anonymization," *Proc. ASJ, Spring meeting,* 3-P-31, pp. 1159–1162, Mar. 2020. (in Japanese)
5. Shunsuke Goto, Kotaro Ohnishi, **Yuki Saito**, Kentaro Tachibana, and Koichiro Mori, "Multi-speaker text-to-speech synthesis using an embedding vector based on a face image," *Proc. ASJ, Spring meeting,* 2-Q-49, pp. 1141–1144, Mar. 2020. (in Japanese)
6. Takaaki Saeki, **Yuki Saito**, Shinnosuke Takamichi, and Hiroshi Saruwatari, "Sub-band lifter-training method for full-band voice conversion using spectral differentials," *Proc. ASJ, Spring meeting,* 2-2-5, pp. 1085–1088, Mar. 2020. (in Japanese)
7. Shinnosuke Takamichi, **Yuki Saito**, Tomohiko Nakamura, Tomoki Koriyama, and Hiroshi Saruwatari, "manga2voice: speech analysis towards audio synthesis from comic image," *Proc. ASJ, Spring meeting,* 1-2-15, pp. 1065–1068, Mar. 2020. (in Japanese)
8. **Yuki Saito**, Shinnosuke Takamichi, and Hiroshi Saruwatari, "SMASH corpus: spontaneous speech corpus of audio commentaries on gameplay," *Proc. ASJ, Spring meeting,* 1-2-14, pp. 1061–1064, Mar. 2020. (in Japanese)
9. Satoshi Naitou, **Yuki Saito**, Shinnosuke Takamichi, Yasuyuki Saito, and Hiroshi Saruwatari, "Estimation of breath position for VOCALOID song sung by user," *Proc. ASJ, Spring meeting,* 1-2-12, pp. 1057–1058, Mar. 2020. (in Japanese)

10. Takaaki Saeki, **Yuki Saito**, Shinnosuke Takamichi, and Hiroshi Saruwatari, "Filter estimation for computational complexity reduction of DNN-based voice conversion using spectral differentials," *Proc. ASJ, Autumn meeting,* 2-4-1, pp. 961–962, Sep. 2019. (in Japanese)
11. Hiroki Tamaru, **Yuki Saito**, Shinnosuke Takamichi, Tomoki Koriyama, and Hiroshi Saruwatari, "Neural double-tracking based on generative moment matching network for users' singing," *Proc. ASJ, Autumn meeting,* 1-4-2, pp. 935–938, Sep. 2019. (in Japanese)
12. Hiroki Tamaru, **Yuki Saito**, Shinnosuke Takamichi, Tomoki Koriyama, and Hiroshi Saruwatari, "Generative moment matching network-based random modulation post-filter for singing voices and its application to double-tracking," *Proc. ASJ, Spring meeting,* 2-10-5, pp. 1035–1038, Mar. 2019. (in Japanese)
13. Satoshi Mizoguchi, **Yuki Saito**, Shinnosuke Takamichi, and Hiroshi Saruwatari, "Low-musical-noise DNN-based speech enhancement applied to noise with various kurtosis," *Proc. ASJ, Spring meeting,* 1-6-6, pp. 185–188, Mar. 2019. (in Japanese)
14. Shinnosuke Takamichi, **Yuki Saito**, Norihiro Takamune, Daichi Kitamura, and Hiroshi Saruwatari, "Phase reconstruction from amplitude spectrograms based on directional-statistics DNNs," *Proc. ASJ, Autumn meeting,* 2-4-2, pp. 1127–1130, Sep. 2018. (in Japanese)
15. Satoshi Mizoguchi, **Yuki Saito**, Shinnosuke Takamichi, and Hiroshi Saruwatari, "Low-musical-noise speech enhancement based on DNNs and kurtosis matching," *Proc. ASJ, Autumn meeting,* 2-1-7, pp. 177–180, Sep. 2018. (in Japanese)
16. 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,* 3-8-8, pp. 243–244, Mar. 2018. (in Japanese)
17. 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)
18. 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)

Co-authors' Awards

1. IPSJ SIG-MUS/SLP Student Poster Award, Jun. 2020. (Awardee: Kazuki Fujii)
2. FujiSankei Business i Awards, Jun. 2020. (Awardee: Kazuki Fujii)
3. IPSJ Yamashita SIG Research Award, Mar. 2020. (Awardee: Shinnosuke Takamichi)
4. The 3rd IEEE Signal Processing Society Tokyo Joint Chapter Student Award, Dec. 2019. (Awardee: Hiroki Tamaru)
5. The 18th Best Student Presentation Award of ASJ, Mar. 2019. (Awardee: Satoshi Mizoguchi)
6. IPSJ SIG-MUS/SLP Presentation Award, Jun. 2018. (Awardee: Shinnosuke Takamichi)

Misc.

1. Four patent applications
2. One copyrighted work (Japanese Versatile Speech: JVS Corpus)

Bibliography

[1] 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.

[2] 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.

[3] A. W. Black, "Speech synthesis for educational technology," in *Proc. SLaTE Workshop*, Farmington, U.S.A., Oct. 2007, pp. 104–107.

[4] Y. Oshima, S. Takamichi, T. Toda, G. Neubig, S. Sakti, and Satoshi Nakamura, "Non-native speech synthesis preserving speaker individuality based on partial correction of prosodic and phonetic characteristics," in *Proc. INTERSPEECH*, Dresden, Germany, Sep. 2015, pp. 299–303.

[5] J. A. Liao, N. Jincho, and H. Kikuchi, "Interactive virtual reality speech simulation system using autonomous audience with natural non-verbal behavior," *International Journal of Machine Learning and Computing*, vol. 8, no. 4, pp. 404–407, Aug. 2018.

[6] M. Macon, L. A. Jensen-Link, J. Oliverio, M. A Clements, and E. George, "A singing voice synthesis system based on sinusoidal modeling," in *Proc. ICASSP*, Munich, Germany, Apr. 1997, pp. 435–438.

[7] 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*. Hollywood, U.S.A., Nov. 2012.

[8] 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.

[9] 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.

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

[11] A. B. Kain, J.-P. Hosom, X. Niu, J. P. H. van Santen, M. Fried-Oken, and J. Staehely, "Improving the intelligibility of dysarthric speech," *Speech Communication*, vol. 49, no. 9, pp. 743–756, Sep. 2007.

[12] K. Nakamura, T. Toda, H. Saruwatari, and K. Shikano, "Speaking-aid systems using GMM-based voice conversion for electrolaryngeal speech," *Speech Communication*, vol. 54, no. 1, pp. 134–146, Jan. 2012.

[13] D .H. Klatt, "Software for a cascade/parallel formant synthesizer," *Journal of the Acoustical Society of America*, vol. 67, no. 3, pp. 971–995, 1980.

[14] A. J. Hunt and A.W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *Proc. ICASSP*, Atlanta, U.S.A., May 1996, pp. 373–376.

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

[16] M. Abe, S. Nakamura, K. Shikano, and H. Kuwabara, "Voice conversion through vector quantization," in *Proc. ICASSP*, New York, U.S.A., Apr. 1988, pp. 655–658.

[17] J. Yamagishi and T. Kobayashi, "Average-voice-based speech synthesis using HSMM-based speaker adaptation and adaptive training," *IEICE Transactions on Information and Systems*, vol. E90-D, no. 2, pp. 533–543, Feb. 2007.

[18] Y. Ohtani, T. Toda, H. Saruwatari, and K. Shikano, "Many-to-many eigenvoice conversion with reference voice," in *Proc. INTERSPEECH*, Brighton, U.K., Sep. 2009, pp. 1623–1626.

[19] F. Fang, X. Wang, J. Yamagishi, I. Echizen, M. Todisco, N. Evans, and J.-F. Bonastre, "Speaker anonymization using x-vector and neural waveform models," in *Proc. SSW*, Vienna, Austria, Sep. 2019, pp. 155–160.

[20] S. Ueno, M. Mimura, S. Sakai, and T. Kawahara, "Multi-speaker sequence-to-sequence speech synthesis for data augmentation in acoustic-to-word speech recognition," in *Proc. ICASSP*, Brighton, U.K., May 2019, pp. 6161–6165.

[21] 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.

[22] 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.

[23] G. E. 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," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, Oct. 2012.

[24] 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.

[25] 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.

[26] O. Abdel-Hamid, A. r. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 10, pp. 1533–1545, Oct. 2014.

[27] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *Proc. ICASSP*, Florence, Italy, May 2014, pp. 4080–4084.

[28] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *Proc. ICASSP*, Alberta, Canada, Apr. 2018, pp. 5329–5333.

[29] 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, Apr. 2016.

[30] D. Rumelhart, G. E. Hinton, and R. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 1, pp. 533–536, Oct. 1986.

[31] 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.

[32] 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.

[33] S. Takamichi, T. Toda, A. W. Black, G. Neubig, S. Sakti, and S. Nakamura, "Postfil-

ters to modify the modulation spectrum for statistical parametric speech synthesis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 4, pp. 755–767, Apr. 2016.

[34] 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.

[35] 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.

[36] 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.

[37] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," *arXiv*, vol. abs/1312.6114, 2013.

[38] C.-C. Hsu, H.-T. Hwang, Y.-C. Wu, Y. Tsao, and H.-M. Wang, "Voice conversion from non-parallel corpora using variational auto-encoder," in *Proc. APSIPA ASC*, Jeju, South Korea, Dec. 2016.

[39] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio, "Generating sentences from a continuous space," in *Proc. CoNLL*, Berlin, Germany, Aug. 2016, pp. 10–21.

[40] N. Dilokthanakul, P. A. M. Mediano, M. Garnelo, M. C. H. Lee, H. Salimbeni, K. Arulkumaran, and M. Shanahan, "Deep unsupervised clustering with gaussian mixture variational autoencoders," *arXiv*, vol. abs/1611.02648, 2016.

[41] N. Hojo, Y. Ijima, and H. Mizuno, "DNN-based speech synthesis using speaker codes," *IEICE Transactions on Information and Systems*, vol. E101-D, no. 2, pp. 462–472, Feb. 2018.

[42] 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.

[43] G. Fant, *Acoustic theory of speech production with calculations based on X-ray studies of Russian articulations*, De Gruyter, 1970.

[44] H. Kawahara, J. 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 *Proc. MAVEBA*, Florence, Italy, Sep. 2001, pp. 59–64.

[45] 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.

[46] 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, Jul. 2011.

[47] 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.

[48] 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, Jul. 2016.

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

[50] 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.

[51] 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.

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

[53] 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.

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

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

[56] 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.

[57] K. Kobayashi, T. Toda, and S. Nakamura, "Intra-gender statistical singing voice conversion with direct waveform modification using log-spectral differential," *Speech*

Communication, vol. 99, pp. 211–220, May 2018.

[58] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, "Semi-supervised learning with deep generative models," in *Proc. NIPS*, Montreal, Canada, Dec. 2014, pp. 3581–3589.

[59] 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.

[60] 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.

[61] 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, Jun. 2000, pp. 1315–1318.

[62] 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, May 2007.

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

[64] 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.

[65] 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.

[66] 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.

[67] 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.

[68] 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.

[69] 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, Oct. 2012.

[70] 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, Jan. 2005.

[71] 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.

[72] 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.

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

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

[75] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. ICML*, Sydney, Australia, Aug. 2017, pp. 214–223.

[76] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *Proc. ICCV*, Venice, Italy, Oct. 2017, pp. 2794–2802.

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

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

[79] 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.

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

[81] 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.

[82] 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.

[83] 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.

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

[85] 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.

[86] Y. Li, S. Kevin, and Z. Richard, "Generative moment matching networks," in *Proc. ICML*, Lille, France, Jul. 2015, pp. 1718–1727.

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

[88] 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, Dec. 2011.

[89] 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.

[90] 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.

[91] 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.

[92] H. Zhang, T. Xu, H. Li, S. Zhang, X. Huang, X. Wang, and D. Metaxas, "Stack-GAN: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *Proc. ICCV*, Venice, Italy, Oct. 2017, pp. 5907–5915.

[93] 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, Feb. 2016.

[94] T. Kaneko, H. Kameoka, N. Hojo, Y. Ijima, K. Hiramatsu, and K. Kashino, "Gen-

erative adversarial network-based postfilter for statistical parametric speech synthesis," in *Proc. ICASSP*, New Orleans, U.S.A., Mar. 2017, pp. 4910–4914.

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

[96] 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.

[97] 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.

[98] 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.

[99] B. Poole, A. Alemi, J. Sohl-dickstein, and A. Angelova, "Improved generator objectives for GANs," *arXiv*, vol. abs/1612.02780, 2016.

[100] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 7–19, Jan. 2015.

[101] Q. Wang, W. Rao, S. Sun, L. Xie, E. S. Chng, and H. Li, "Unsupervised domain adaptation via domain adversarial training for speaker recognition," in *Proc. ICASSP*, Alberta, Canada, Apr. 2018, pp. 4889–4893.

[102] K. Tokuda, T. Kobayashi, T. Fukada, H. Saito, and S. Imai, "Spectral estimation of speech based on mel-cepstral representation," *Transactions on IEICE*, vol. J74-A, no. 8, pp. 1240–1248, Aug. 1991.

[103] 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.

[104] L. Juvela, B. Bollepalli, X. Wang, H. Kameoka, M. Airaksinen, J. Yamagishi, and P. Alku, "Speech waveform synthesis from MFCC sequences with generative adversarial networks," in *Proc. ICASSP*, Calgary, Canada, Apr. 2018, pp. 5679–5683.

[105] C. Donahue J. McAuley and M. Puckette, "Synthesizing audio with GANs," in *Proc. ICLR Workshop*, Vancouver, Canada, May 2018.

[106] R. Sonobe, S. Takamichi, and H. Saruwatari, "JSUT corpus: Free large-scale Japanese speech corpus for end-to-end speech synthesis," *arXiv*, vol.

abs/1711.00354, 2017.

[107] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *Proc. ICML*, Sydney, Australia, Aug. 2017, pp. 933–941.

[108] T. Kaneko and H. Kameoka, "CycleGAN-VC: Non-parallel voice conversion using cycle-consistent adversarial networks," in *Proc. EUSIPCO*, Rome, Italy, Sep. 2018, pp. 2114–2118.

[109] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "StarGAN-VC: Non-parallel many-to-many voice conversion using star generative adversarial networks," in *Proc. SLT*, Greece, Athens, Dec. 2018, pp. 266–273.

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

[111] L. Sun, K. Li, H. Wang, S. Kang, and H. Meng, "Phonetic posteriorgrams for many-to-one voice conversion without parallel data training," in *Proc. ICME*, Seattle, U.S.A., Jul. 2016.

[112] M. Sambur, "Selection of acoustic features for speaker identification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 2, pp. 176–182, Apr. 1975.

[113] Y. Zhang, M. Pezeshki, P. Brakel, S. Zhang, C. L. Y. Bengio, and A. Courville, "Towards end-to-end speech recognition with deep convolutional neural networks," in *Proc. INTERSPEECH*, California, U.S.A., Sep. 2016, pp. 410–414.

[114] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in *Proc. ICASSP*, Shanghai, China, Mar. 2016, pp. 5115–5119.

[115] A. Tjandra, S. Sakti, and S. Nakamura, "Listening while speaking: Speech chain by deep learning," in *Proc. ASRU*, Okinawa, Japan, Dec. 2017, pp. 301–308.

[116] H. Miyoshi, Y. Saito, S. Takamichi, and H. Saruwatari, "Voice conversion using sequence-to-sequence learning of context posterior probabilities," in *Proc. INTERSPEECH*, Stockholm, Sweden, Aug. 2017, pp. 1268–1272.

[117] M. Miiller, S. Stiiker, and A. Waibel, "Multilingual adapataion of RNN-based ASR systems," in *Proc. ICASSP*, Alberta, Canada, Apr. 2018, pp. 5219–5223.

[118] H.-T. Luong, S. Takaki, G. E. Henter, and J. Yamagishi, "Adapting and controlling DNN-based speech synthesis using input codes," in *Proc. ICASSP*, New Orleans, U.S.A., Mar. 2017, pp. 1905–1909.

[119] M. Wolfel, "Channel selection by class separability measures for automatic transcriptions on distant microphones," in *Proc. INTERSPEECH*, Antwerp, Belgium,

Aug. 2007, pp. 582–585.

[120] E. Cooper, C.-I Lai, Y. Yasuda, F. Fang, X. Wang, N. Chen, and J. Yamagishi, "Zero-shot multi-speaker text-to-speech with state-of-the-art neural speaker embeddings," in *Proc. ICASSP*, Barcelona, Spain, May 2020, pp. 6184–6188.

[121] A. J. Quinn and B. B. Bederson, "Human computation: A survey and taxonomy of a growing field," in *Proc. SIGCHI*, Vancouver, Canada, May 2011, pp. 1403–1412.

[122] J. Li, Y. Baba, and H. Kashima, "Simultaneous clustering and ranking from pairwise comparisons," in *Proc. IJCAI*, Stockholm, Sweden, Jul. 2018, pp. 1554–1560.

[123] B. Settles, "Active learning literature survey," Computer Sciences Technical Report 1648, University of Wisconsin–Madison, Jan. 2009.

[124] M. Tachibana, T. Nose, J. Yamagishi, and T. Kobayashi, "A technique for controlling voice quality of synthetic speech using multiple regression HSMM," in *Proc. ICSLP*, Pittsburgh, U.S.A., Sep. 2006, pp. 2438–2441.

[125] K. Ohta, Y. Ohtani, T. Toda, H. Saruwatari, and K. Shikano, "Regression approaches to voice quality control based on one-to-many eigenvoice conversion," in *Proc. ICSLP*, Bonn, Germany, Aug. 2007, pp. 101–106.

[126] J. Lorenzo-Trueba, G. E. Henter, S. Takaki, J. Yamagishi, Y. Morino, and Y. Ochiai, "Investigating different representations for modeling and controlling multiple emotions in DNN-based speech synthesis," *Speech Communication*, vol. 99, pp. 135–143, May 2018.

[127] J. R. Hershey, Z. Chen, J. L. Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Proc. ICASSP*, Shanghai, China, Mar. 2016, pp. 31–35.

[128] P. Goyal and E. Ferrara, "Graph embedding techniques, applications, and performance: A survey," *Knowledge-Based Systems*, vol. 151, pp. 78–94, Jul. 2018.

[129] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, May 2013.

[130] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning System*, Mar. 2020, (Early Access).

[131] F. M. Zanzotto, "Viewpoint: Human-in-the-loop artificial intelligence," *Journal of Artificial Intelligence Research*, vol. 64, pp. 243–252, Feb. 2019.

[132] K. Fujii, Y. Saito, S. Takamichi, Y. Baba, and H. Saruwatari, "HumanGAN: Gen-

erative adversarial network with human-based discriminator and its evaluation in speech perception modeling," in *Proc. ICASSP*, Barcelona, Spain, May 2020, pp. 6239–6243.

[133] K. Itou, M. Yamamoto, K.Takeda, T. Takezawa, T. Matsuoka, T. Kobayashi, K. Shikano, and S. Itahashi, "JNAS: Japanese speech corpus for large vocabulary continuous speech recognition research," *Journal of the Acoustical Society of Japan (E)*, vol. 20, no. 3, pp. 199–206, May 1999.

[134] C. D. Brown and H. T. Davis, "Receiver operating characteristics curves and related decision measures: A tutorial," *Chemometrics and Intelligent Laboratory Systems*, vol. 80, no. 1, pp. 24–38, Jan. 2006.

[135] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve," *Radiology*, vol. 143, no. 1, pp. 29–36, Apr. 1982.

[136] N. Iwahashi and Y. Sagisaka, "Speech spectrum conversion based on speaker interpolation and multi-functional representation with weighting by radial basis function networks," *Speech Communication*, vol. 16, no. 2, pp. 139–151, Feb. 1995.

[137] K. Akuzawa, Y. Iwasawa, and Y. Matsuo, "Expressive speech synthesis via modeling expressions with variational autoencoder," in *Proc. INTERSPEECH*, Hyderabad, India, Sep. 2018, pp. 3067–3071.

[138] S. O. Arik, J. Chen, K. Peng, W. Ping, and Y. Zhou, "Neural voice cloning with a few samples," in *Proc. NeurIPS*, Montreal, Canada, Dec. 2018, pp. 10019–10029.

[139] A. Tamamori, T. Hayashi, K. Kobayashi, K. Takeda, and T. Toda, "Speaker-dependent WaveNet vocoder," in *Proc. INTERSPEECH*, Stockholm, Sweden, Aug. 2017, pp. 1118–1122.

[140] T. Baumann, "Partial representations improve the prosody of incremental speech synthesis," in *Proc. INTERSPEECH*, Singapore, Sep. 2014, pp. 2932–2936.

[141] R. Arakawa, S. Takamichi, and H. Saruwatari, "Implementation of DNN-based real-time voice conversion and its improvements by audio data augmentation and mask-shaped device," in *Proc. SSW*, Vienna, Austria, Sep. 2019, pp. 93–98.

[142] T. Saeki, Y. Saito, S. Takamichi, and H. Saruwatari, "Lifter training and sub-band modeling for computationally efficient and high-quality voice conversion using spectral differentials," in *Proc. ICASSP*, Barcelona, Spain, May 2020, pp. 7784–7788.

[143] B. Li and H. Zen, "Multi-language multi-speaker acoustic modeling for LSTM-RNN based statistical parametric speech synthesis," in *Proc. INTERSPEECH*, California,

U.S.A., Sep. 2016, pp. 2468–2472.

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

[145] X. Wang, S. Takaki, and J. Yamagishi, "Investigating very deep highway networks for parametric speech synthesis," in *Proc. SSW*, California, U.S.A., Sep. 2016, pp. 166–171.

[146] 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.

[147] 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.

[148] 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.

[149] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *Journal of Machine Learning Research*, vol. 17, no. 59, pp. 1–35, Apr. 2016.

[150] S. Sun, C.-F. Yeh, M.-Y. Hwang, M. Ostendorf, and L. Xie, "Domain adversarial training for accented speech recognition," in *Proc. ICASSP*, Alberta, Canada, Apr. 2018, pp. 4854–4858.

[151] J.-C. Chou, C.-C. Yeh, H.-Y. Lee, and L.-S. Lee, "Multi-target voice conversion without parallel data by adversarially learning disentangled audio representations," in *Proc. INTERSPEECH*, Hyderabad, India, Sep. 2018, pp. 501–505.

[152] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "CycleGAN-VC2: Improved CycleGAN-based non-parallel voice conversion," in *Proc. ICASSP*, Brighton, U.K., May 2019, pp. 6820–6824.

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

[154] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "StarGAN-VC2: Rethinking conditional methods for stargan-based voice conversion," in *Proc. INTERSPEECH*, Graz, Austria, Sep. 2019, pp. 679–683.

[155] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation," in

Prop. CVPR, Salt Lake City, U.S.A., Jun. 2018, pp. 8789–8797.

[156] M. Mirza and S. Osindero, "Conditional generative adversarial networks," *arXiv*, vol. abs/1411.1784, 2014.

[157] Y. Zhou, X. Tian, H. Xu, R. K. Das, and H. Li, "Cross-lingual voice conversion with bilingual phonetic posteriorgram and average modeling," in *Proc. ICASSP*, Brighton, U.K., May 2019, pp. 6790–6794.

[158] S. H. Mohammadi and T. Kim, "One-shot voice conversion with disentangled representations by leveraging phonetic posteriorgrams," in *Proc. INTERSPEECH*, Graz, Austria, Sep. 2019, pp. 704–708.

[159] 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.

[160] H. Lu, Z. Wu, R. Li, S. Kang, J. Jia, and Helen Meng, "A compact framework for voice conversion using wavenet conditioned on phonetic posteriorgrams," in *Proc. ICASSP*, Brighton, U.K., May 2019, pp. 6810–6814.

[161] S. Liu, Y. Cao, X. Wu, L. Sun, X. Liu, and H. Meng, "Jointly trained conversion model and WaveNet vocoder for non-parallel voice conversion using mel-spectrograms and phonetic posteriorgrams," in *Proc. INTERSPEECH*, Graz, Austria, Sep. 2019, pp. 704–708.

[162] K. Sugiura, Y. Shiga, H. Kawai, T. Misu, and C. Hori, "A cloud robotics approach towards dialogue-oriented robot speech," *Advanced Robotics*, vol. 29, no. 7, pp. 449–456, Mar. 2015.

[163] K. Maekawa, H. Koiso, S. Furui, and H. Isahara, "Spontaneous speech corpus of Japanese," in *Proc. LREC*, Athens, Greece, May 2000, pp. 947–952.

[164] A. Camacho, "Swipe: A sawtooth waveform inspired pitch estimator for speech and music," *Ph.D. dissertation, University of Florida*, 2007.

[165] D. Talkin, "A robust algorithm for pitch tracking (RAPT)," *Speech Coding and Synthesis*, pp. 495–518, 1995.

[166] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, Atlanta, U.S.A., Jun. 2013.

[167] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, Apr. 2014.

[168] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. ICML*, Lille, France, Jul. 2015, pp.

448–456.

[169] B. W. Matthews, "Comparison of the predicted and observed secondary structure of t4 phage lysozyme," *Biochimica et Biophysica Acta (BBA) - Protein Structure*, vol. 405, no. 2, pp. 442–451, Oct. 1975.

[170] A. Kurematsu, K. Takeda, Y. Sagisaka, S. Katagiri, H. Kuwabara, and K. Shikano, "ATR Japanese speech database as a tool of speech recognition and synthesis," *Speech Communication*, vol. 9, no. 4, pp. 357–363, Aug. 1990.

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 through my master's course and doctoral course. I have learned many valuable aspects of being a researcher from his attitude toward research.

I would also like to express my appreciation to Professor Masahiko Inami, Professor Tatsuya Harada, Associate Professor Masaaki Kondo, and Lecturer at MI center Tomoki Koriyama of the University of Tokyo, members of the thesis committee, for their valuable comments and suggestions on this thesis.

I would especially like to express my gratitude to Research Associate 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 past Visiting Professor Kunio Kashino (currently Senior Distinguished Researcher of NTT Communication Science Laboratories), Lecturer Shoichi Koyama, past Research Assistant Professor Daichi Kitamura (currently Research Associate of National Institute of Technology, Kagawa College), Research Assistant Professor Tomohiko Nakamura, and Academic Support Specialist Norihiro Takamune of the University of Tokyo, for their technical comments and lessons.

I want to 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.

I experienced many research internship through my master's course and doctoral course. I would like to thank my research mentors during the internship, Senior Distinguished Researcher Hirokazu Kameoka of NTT Communication Science Laboratories, Distinguished Researchers Yusuke Ijima, Kyosuke Nishida, and Ryo Masumura of NTT Media Intelligence Laboratories, and Research Engineer Kentaro Tachibana of LINE Corp. (past Research Engineer of DeNA Corp.), for their consistent supports. I would like to offer my special thanks to Professor Tomoki Toda of Nagoya University, Professor Junichi Yamag-

ishi of National Institute of Informatics, and Senior Staff Research Scientist Heiga Zen of Google Research, for their insightful comments to my work related to this thesis.

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

Appendix A

VC Using Input-to-output Highway Networks

A.1 Introduction

In acoustic modeling for VC, not only techniques to alleviate over-smoothing, but also input speech information can be used since the input and output parameters are often in the same domain (e.g., cepstrum). This appendix proposes a DNN-based VC method using input-to-output highway networks. The proposed method generates spectral parameters of converted speech as the sum of input spectral parameters and weighted spectral differentials predicted by DNNs. The use of input speech parameters effectively alleviates over-smoothing, and the weights of the spectral differentials control which components of input speech parameters should be converted by the DNNs.

A.2 Proposed VC Method

A.2.1 Input-to-Output Highway Networks for VC

Highway networks [144, 145] 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, this section proposes a VC method using input-to-output highway networks that have three components: skip connection, transform gate $\boldsymbol{T}(\cdot)$, and spectral differentials estimator $\boldsymbol{G}(\cdot)$. The converted speech parameters with this method are generated as follows:

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

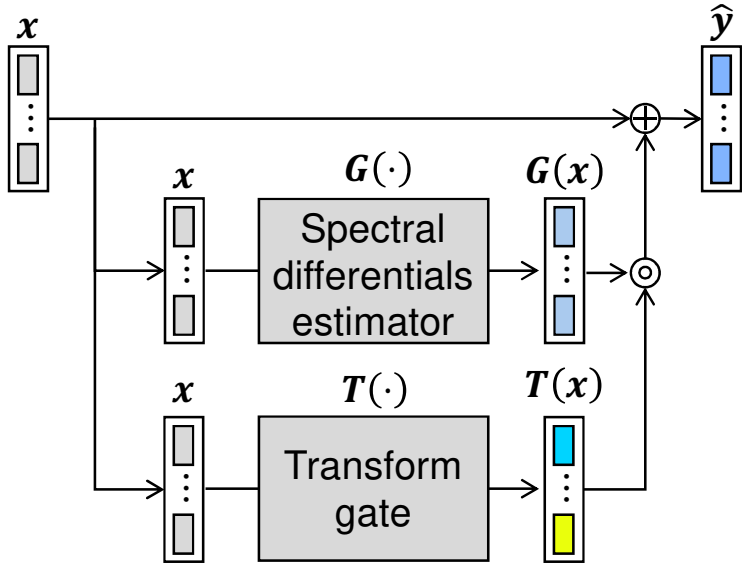

Fig. A.1. VC using input-to-output highway networks

The transform gate $\boldsymbol{T}(\cdot)$ is described as Feed-Forward DNNs. Each value of $\boldsymbol{T}(\boldsymbol{x})$ ranges from 0.0 to 1.0, representing time- and feature-varying weights of spectral differentials predicted as $\boldsymbol{G}(\boldsymbol{x})$. 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 [110]. Therefore, input speech parameters are strongly transformed by $\boldsymbol{G}(\cdot)$ when the transform gate's value becomes close to 1.0. Figure A.1 shows the proposed VC method using input-to-output highway networks. The loss function for training DNNs is equal to the MGE loss shown in Eq. (2.16). Model parameters of $\boldsymbol{T}(\cdot)$ and $\boldsymbol{G}(\cdot)$ are simultaneously estimated to minimize the loss function.

A.2.2 Discussions

Since the proposed method utilizes both input speech parameters and spectral differentials weighted by the transform gate, it efficiently alleviates over-smoothing of the converted speech parameters. Figure 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., close to use of residual networks) or small (i.e., close to direct use of input speech parameters). The proposed input-to-output highway networks alleviate over-smoothing better than conventional Feed-forward DNNs in both cases.

The variation in spectral parameters between speakers strongly depends not only on speaker pairs, but also on frequency bands and phonetic environments. For example,

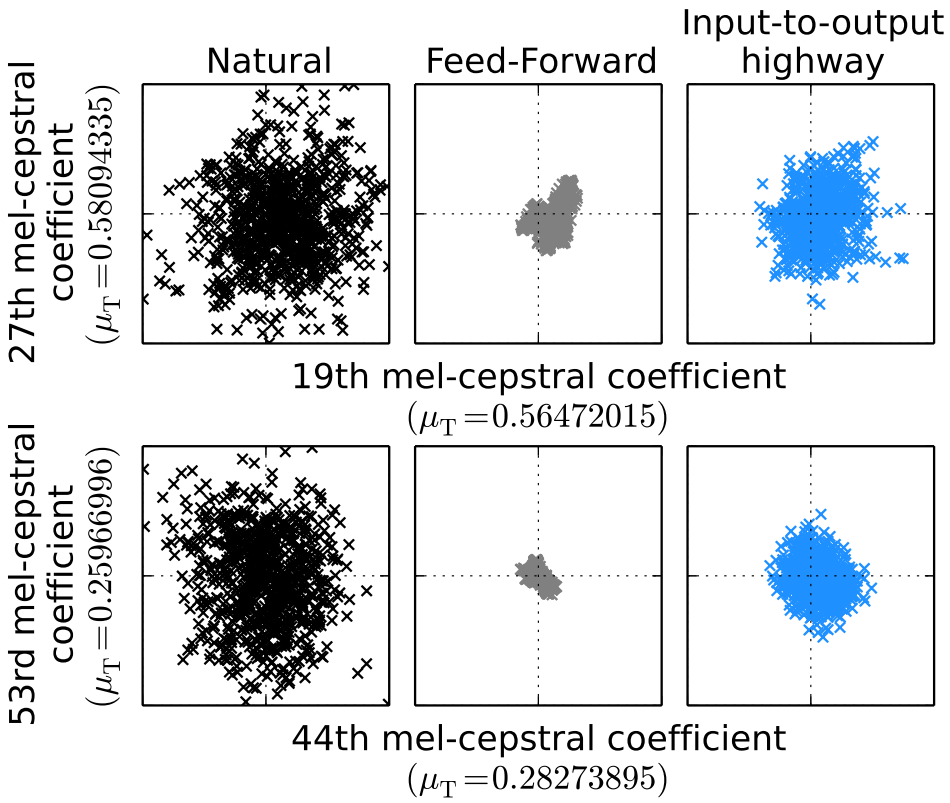

Fig. A.2. Scatter plots of speech parameters. μ_T denotes transform gate's value averaged over one utterance.

formant structure differs greatly between the genders, and the inter-speaker variation in low-frequency band is small within the same gender. On the other hand, the inter-phoneme variation (i.e., intra-speaker variation) in the low frequency band tends to be large [146]. Therefore, a *golden* VC method is desirable to have an acoustic model that avoids excessive conversion when the difference between input and output features is small (e.g., frequency warping [147]) and provides flexible conversion when the difference is large. The proposed method can achieve such VC by using the spectral differential estimation learned with data-driven weights. Figure A.3 shows examples of transform gate's values predicted from mel-filter banks. This figure shows that $\boldsymbol{G}(\cdot)$ greatly transforms the spectral parameters in the high frequency band, since they strongly represent voice characteristics of a speaker. Meanwhile, in male-to-male VC (Fig. A.3(a)), $\boldsymbol{G}(\cdot)$ does not transform the spectral parameters in the low frequency band unlike that of male-to-female VC (Fig. A.3(b)).

From another perspective, the transform gate in the proposed method can be regarded as *soft* selection of features. The dimensionality of spectral features (e.g., the order of

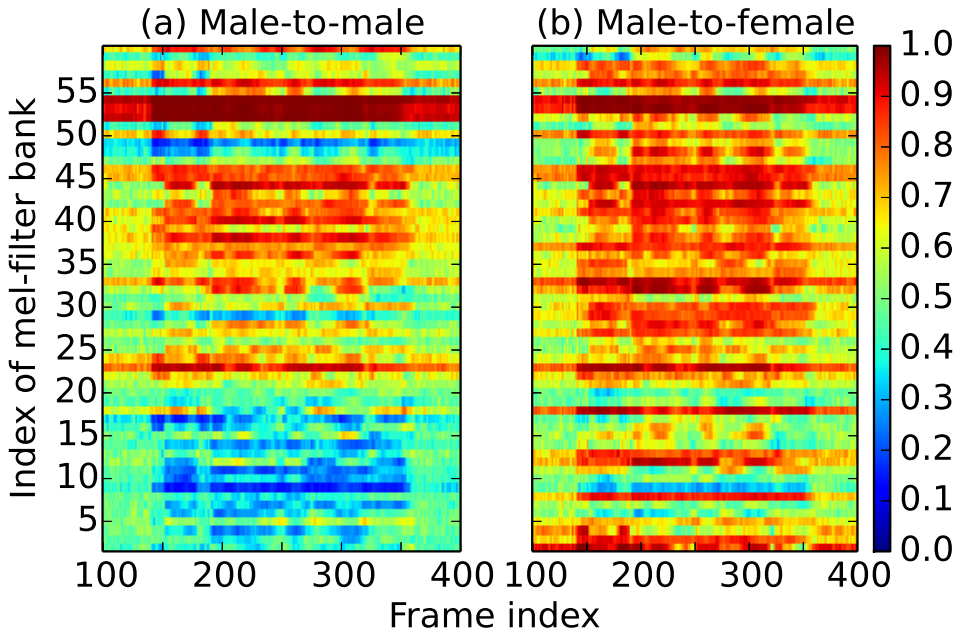

Fig. A.3. Examples of transform gate's values predicted from mel-filter banks

mel-cepstral coefficients) is a hyperparameter for VC. For instance, the use of only lower order of mel-cepstra makes the acoustic model training robust while it degrades speech quality due to insufficient conversion of spectral features. On the other hand, the use of higher order improves speech quality but suffers from the randomness of the spectral features. The former case corresponds to $\boldsymbol{T}(\boldsymbol{x}) = \mathbf{1}$ for the lower order and $\boldsymbol{T}(\boldsymbol{x}) = \mathbf{0}$ for the higher order. The latter case corresponds to $\boldsymbol{T}(\boldsymbol{x}) = \mathbf{1}$ for all orders. Whereas such a *hard* selection is often used, the proposed method can utilize a *soft* selection by the transform gate. Figure A.4 shows examples of transform gate's values predicted from mel-cepstral coefficients. Lower orders of mel-cepstral coefficients, which are dominant in speaker conversion, tend to be strongly transformed by $\boldsymbol{G}(\cdot)$. On the other hand, higher orders of mel-cepstral coefficients tend to be not completely ignored, but weakly converted.

The transform gate in the proposed method is similar to adaptive soft-masking [148] in speech enhancement. Hence, it is expected that knowledge can be shared between VC and speech enhancement.

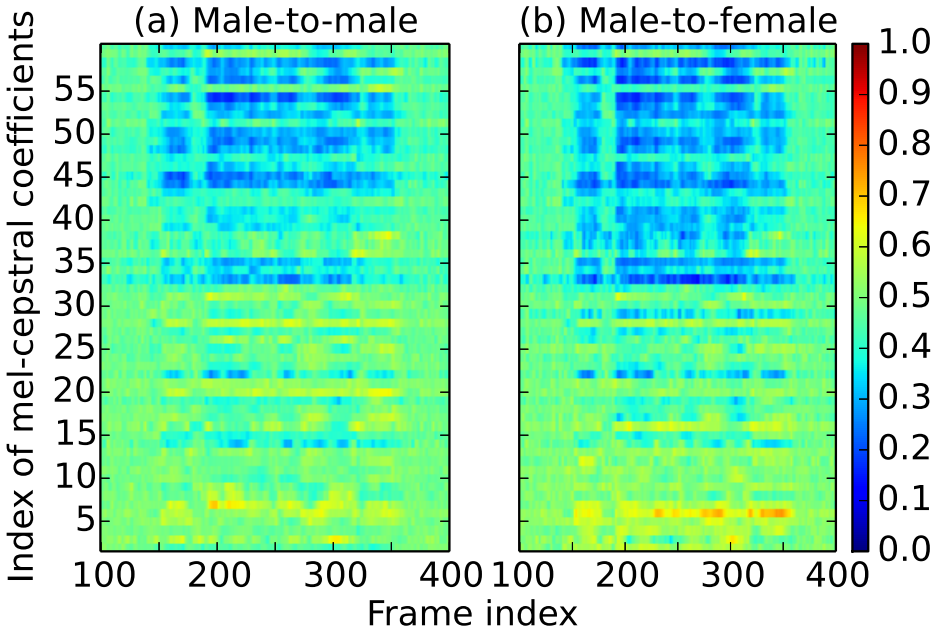

Fig. A.4. Examples of transform gate's values predicted from mel-cepstral coefficients

A.3 Experimental Evaluation

A.3.1 Experimental Conditions

A speech corpus of two male speakers and one female speaker was used. The speakers uttered 503 phonetically balanced sentences [95]. The numbers of sentences for the training and evaluation were 450 (subsets A to I) and 53 (subset J), respectively. The sampling rate of speech signals was 16 kHz. The shift length was set to 5 ms. The 0th-through-59th mel-cepstral coefficients were used as the spectral parameters. F_0 and 5 band-aperiodicity [44, 96] were used as excitation parameters. The STRAIGHT vocoder [47] was used for the parameter extraction and speech waveform synthesis. The 0th mel-cepstral coefficients of input speech were used as those of the converted speech. Speech parameter trajectory smoothing [97] with a 50 Hz cutoff modulation frequency was applied to the spectral parameters in the training data for improving training accuracy. In the training phase, the spectral parameters were normalized to have zero-mean and unit-variance. The MGE training [60] was performed with 25 iterations. Two DNNs for male-to-male and male-to-female VC were trained. The DNN architectures were Feed-Forward networks that included three 512-unit ReLU [54] hidden layers and a 118-unit linear

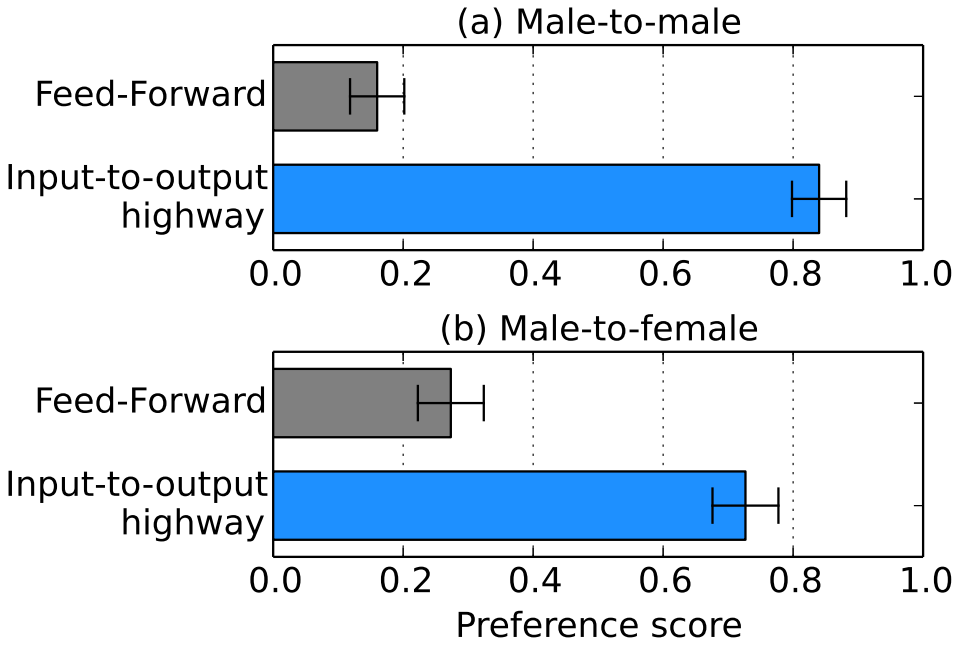

Fig. A.5. Preference scores of converted speech's speech quality with 95% confidence intervals (DNN-based VC using input-to-output highway networks)

output layer. The acoustic model predicted static and dynamic mel-cepstral coefficients (118-dim.) frame by frame. The transform gate had a 59-unit input and 59-unit sigmoid output layers. The optimization algorithm was AdaGrad [98]. The learning rate was set to 0.01. In the VC process, linearly transformed F_0 and source-speaker's band-aperiodicity were used.

A.3.2 Subjective Evaluation

Subjective evaluations on quality and speaker individuality of converted speech were conducted. In the subjective evaluation, the proposed method was compared with a conventional one that uses Feed-Forward DNNs as an acoustic model. A preference AB test on the speech quality was conducted. Every pair of speech samples converted by the two methods was presented in random order. Listeners were asked to select speech samples that sounded like they had better quality. Similarly, a preference XAB test on the speaker individuality was conducted using target speaker's natural speech as the reference "X." Thirty listeners participated in each assessment, using crowdsourced evaluation systems.

Figures A.5 and A.6 show the evaluation results regarding the converted speech quality and speaker individuality, respectively. The proposed method outperforms the con-

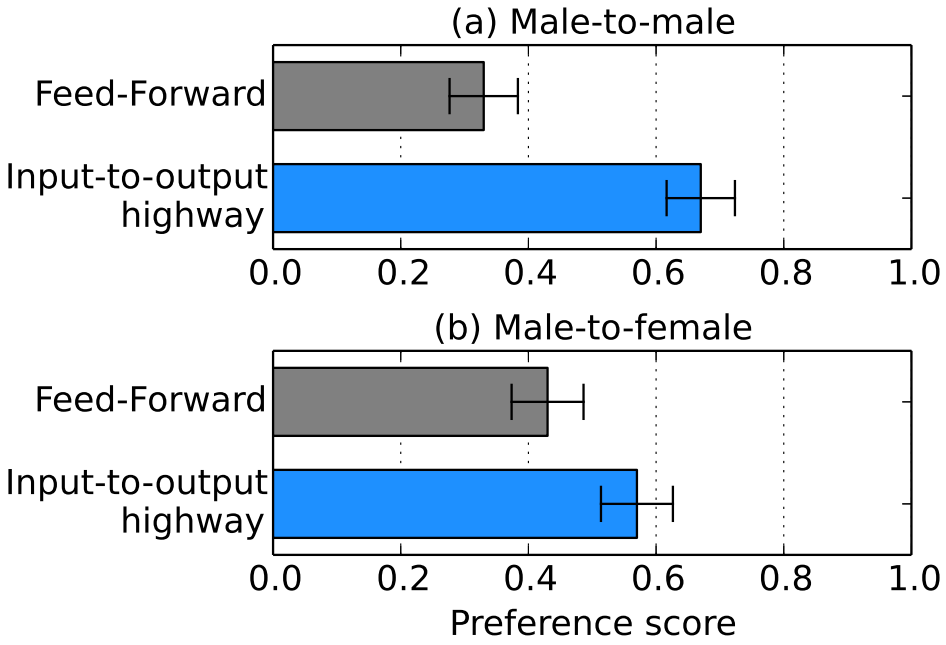

Fig. A.6. Preference scores of converted speech's speaker individuality with 95% confidence intervals (DNN-based VC using input-to-output highway networks)

ventional one regarding both speech quality and speaker individuality. These results demonstrate the effectiveness of the proposed method.

Appendix B

Joint Adversarial Training Algorithm for DNN-based Many-to-one VC

B.1 Introduction

This appendix proposes an adversarial algorithm to train DNNs for many-to-one VC using speech recognition and synthesis. The proposed algorithm incorporates the GAN-based speech synthesis method (Chapter 3) and domain-adversarial training (DAT) of a speech recognition model for improving both converted speech naturalness and speaker similarity. The algorithm jointly trains the speech synthesis and recognition models to optimize them for many-to-one VC.

B.2 Conventional Algorithm for DNN-based Many-to-one VC

The conventional algorithm [111] trains DNN-based speech recognition and synthesis models to represent mapping from any arbitrary input speech to the target speech, using PPGs as the intermediate representation of the VC process.

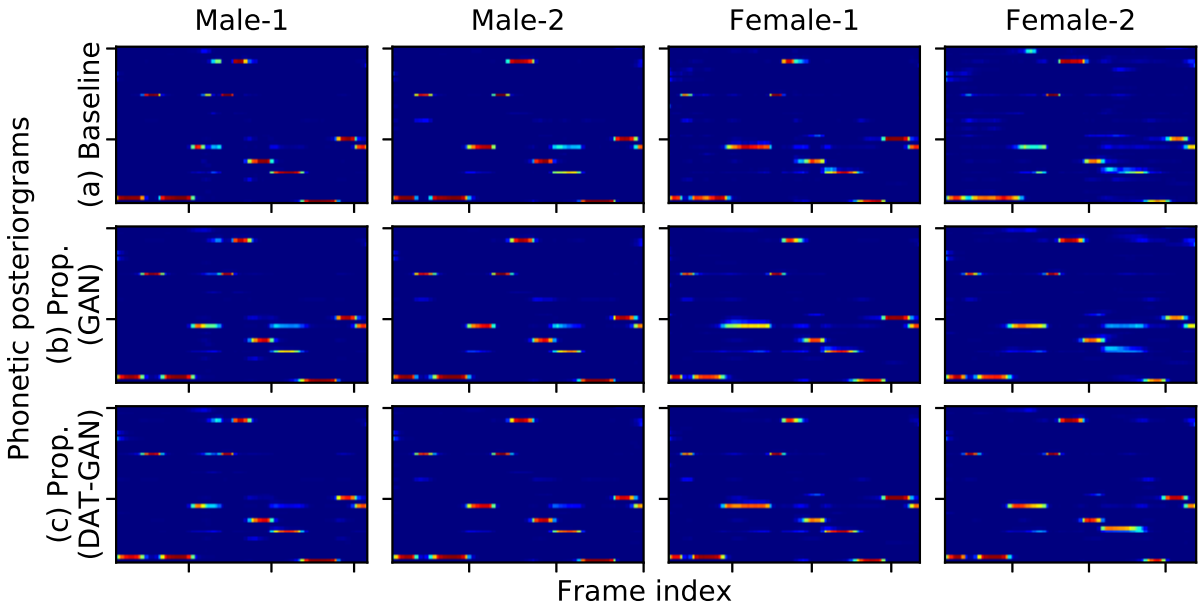

Fig. B.1. Examples of PPGs predicted by speech parameters of four different speakers who uttered same sentences used in subjective evaluation described in section B.4.3. Horizontal and vertical axes represent temporal axis and phoneme index, respectively. Brighter values denote high posterior probabilities. Ranges of temporal axes in these figures were modified for clear illustration.

B.2.1 Speech Recognition Model Training

The recognition model $\boldsymbol{R}(\cdot)$ is trained to predict a phoneme label sequence $\boldsymbol{p}$ from an input speech parameter sequence $\boldsymbol{x}$. A pair of the phoneme label and input speech parameter is sampled from a multi-speaker corpus $\mathcal{D}^{(\mathrm{M})} = \{(\boldsymbol{x}_n^{(\mathrm{M})}, \boldsymbol{p}_n^{(\mathrm{M})})\}_{n=1}^{N^{(\mathrm{M})}}$, where $N^{(\mathrm{M})}$ denotes the amount of training data for the recognition model. A PPG sequence $\hat{\boldsymbol{p}}^{(\mathrm{M})} = R(\boldsymbol{x}^{(\mathrm{M})})$ is predicted by the recognition model. The recognition model $\boldsymbol{R}(\cdot)$ is trained to minimize the phoneme prediction loss defined as the SCE between the phoneme label and PPG, i.e., $L_{\mathrm{SCE}}(\boldsymbol{p}^{(\mathrm{M})}, \hat{\boldsymbol{p}}^{(\mathrm{M})})$.

B.2.2 Speech Synthesis Model Training

Assuming that $\boldsymbol{R}(\cdot)$ is the speaker-independent speech recognition model, the target-speaker-dependent speech synthesis model $\boldsymbol{G}(\cdot)$ is trained to generate a target speech parameter sequence $\boldsymbol{y}$. A pair of the input and target speech parameters is sampled from a target speaker corpus $\mathcal{D}^{(\mathrm{O})} = \{(\boldsymbol{x}_n^{(\mathrm{O})}, \boldsymbol{y}_n^{(\mathrm{O})})\}_{n=1}^{N^{(\mathrm{O})}}$, where $N^{(\mathrm{O})}$ denotes the amount of training data for the synthesis model. A generated speech parameter sequence $\hat{\boldsymbol{y}}^{(\mathrm{O})}$ is

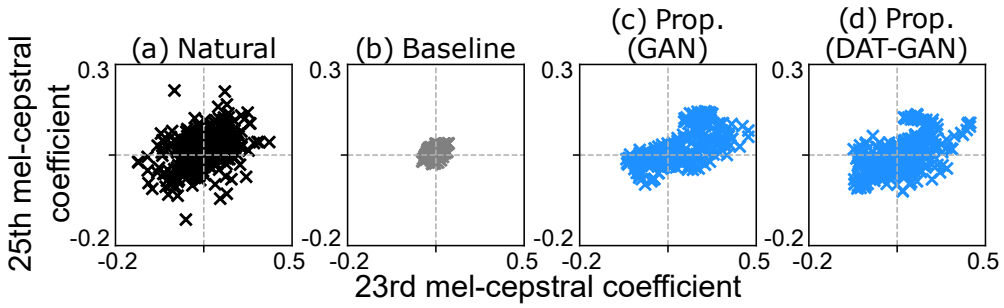

Fig. B.2. Scatter plots mel-cepstral coefficients of (a) natural speech and (b)–(d) converted speech by three different algorithms. Female target speaker's one utterance excluded from training data is used for extracting these mel-cepstral coefficients.

obtained through the recognition and synthesis models, i.e., $\hat{\boldsymbol{y}}^{(\mathrm{O})} = \boldsymbol{G}(\boldsymbol{R}(\boldsymbol{x}^{(\mathrm{O})}))$. The synthesis model $\boldsymbol{G}(\cdot)$ is trained to minimize the MSE between the target and generated speech parameter sequences, i.e., $L_{\mathrm{MSE}}(\boldsymbol{y}^{(\mathrm{O})}, \hat{\boldsymbol{y}}^{(\mathrm{O})})$. Note that model parameters of $\boldsymbol{R}(\cdot)$ are not updated by this training.

B.2.3 Problems

The conventional algorithm can train the DNNs for many-to-one VC without requiring parallel speech corpora. However, actual PPGs fed into the synthesis model can be different among input speakers as shown in Fig. B.2(a). One of the reason is that the speech recognition model training using the phoneme prediction loss does not guarantee to learn speaker-invariant PPGs. The PPG differences can degrade the converted speech quality because the synthesis model trained on only the target speaker's PPGs does not necessarily generalize to any source speakers' PPGs. Moreover, over-smoothing of generated speech parameters shown in Fig. B.1(b) considerably deteriorates the converted speech quality.

B.3 Proposed Algorithm for DNN-based Many-to-one VC Using Adversarial Training

B.3.1 Joint Adversarial Training of Speech Recognition and Synthesis Models

DAT-based Speech Recognition Model Training for Many-to-one VC

The DAT [149] is a general framework to train a DNN-based recognition model that is more robust towards variation in input features by learning domain-invariant latent variables. It has been applied to DNN-based speech information processing such as accented speech recognition [150] and robust speaker classification [101]. Although the DAT was originally invented to improve the recognition model performance, Chou et al. [151] demonstrated its efficacy in autoencoder-based VC that learns speaker-independent latent variables. Note that their method does not guarantee that the latent variables represent phonetic content of input speech, unlike many-to-one VC using PPGs. To improve the converted speech quality of many-to-one VC, it is crucial to obtain PPGs that are invariant to variation in input speakers. Therefore, the DAT in the proposed algorithm regards the two speech corpora used for the DNN training, i.e., 1) the multi-speaker corpus $\mathcal{D}^{(\mathrm{M})}$ and 2) the target speaker corpus $\mathcal{D}^{(\mathrm{O})}$, as the domains. The DAT tries to minimize the difference between the two domains and learn the speaker-invariant speech recognition model that produces better PPGs for many-to-one VC.

For clear formulation, the recognition model $\boldsymbol{R}(\cdot)$ are split into two sub-models as $\boldsymbol{R}(\cdot) = \boldsymbol{R}_{\mathrm{p}}(\boldsymbol{R}_{\mathrm{f}}(\cdot))$. The first sub-model $\boldsymbol{R}_{\mathrm{f}}(\cdot)$ is a feature extractor that extracts features $\hat{\boldsymbol{f}}$ representing phonetic content of input speech from input speech parameters as $\hat{\boldsymbol{f}} = \boldsymbol{R}_{\mathrm{f}}(\boldsymbol{x})$. The second sub-model $\boldsymbol{R}_{\mathrm{p}}(\cdot)$ is a phoneme predictor that predicts PPGs from the extracted features, i.e., $\hat{\boldsymbol{p}} = \boldsymbol{R}_{\mathrm{p}}(\hat{\boldsymbol{f}}) = \boldsymbol{R}_{\mathrm{p}}(\boldsymbol{R}_{\mathrm{f}}(\boldsymbol{x}))$. To capture the domain difference, the DAT introduces a domain classifier $C(\cdot)$ that uses the features $\hat{\boldsymbol{f}}$ to identify a domain where input speech parameters belong to. The domain classifier $C(\cdot)$ is trained to minimize the domain classification loss defined as:

$$L_{\mathrm{dc}}\left(\hat{\boldsymbol{f}}^{(\mathrm{M})}, \hat{\boldsymbol{f}}^{(\mathrm{O})}\right) = -\log C\left(\hat{\boldsymbol{f}}^{(\mathrm{O})}\right) - \log\left(1 - C\left(\hat{\boldsymbol{f}}^{(\mathrm{M})}\right)\right), \tag{B.1}$$

where $\hat{\boldsymbol{f}}^{(\mathrm{M})}$ and $\hat{\boldsymbol{f}}^{(\mathrm{O})}$ are extracted from $\boldsymbol{x}^{(\mathrm{M})}$ and $\boldsymbol{x}^{(\mathrm{O})}$, respectively. On the other hand,

the recognition model $\boldsymbol{R}(\cdot)$ is trained to minimize the loss defined as follows:

$$L_{\mathrm{R}}\left(\boldsymbol{p}^{(\mathrm{M})}, \hat{\boldsymbol{p}}^{(\mathrm{M})}, \hat{\boldsymbol{f}}^{(\mathrm{M})}, \hat{\boldsymbol{f}}^{(\mathrm{O})}\right) = L_{\mathrm{SCE}}\left(\boldsymbol{p}^{(\mathrm{M})}, \hat{\boldsymbol{p}}^{(\mathrm{M})}\right) - \omega_{\mathrm{C}} L_{\mathrm{dc}}\left(\hat{\boldsymbol{f}}^{(\mathrm{M})}, \hat{\boldsymbol{f}}^{(\mathrm{O})}\right), \tag{B.2}$$

where ω_{C} is a hyperparameter that controls the effect of the second term. The loss function can be regarded as the weighted sum of the phoneme prediction loss and the loss to make $C(\cdot)$ misclassify the domains by learning the domain-invariant features $\hat{\boldsymbol{f}}$. Therefore, the minimization of Eq. (B.2) can be expected to reduce the PPG differences among input speakers.

GAN-based Speech Synthesis Model Training for Many-to-one VC

The proposed algorithm also introduces the GAN-based training method for speech synthesis (Chapter 3) to overcome over-smoothing of generated speech parameters. Referring to the subjective evaluation results in Section 3.4.10, W-GAN [75]-based algorithm (Section 3.3.4) was adopted.

Joint Optimization of Speech Recognition and Synthesis Models

To optimize both the recognition and synthesis models for many-to-one VC, the proposed algorithm *jointly* train the two models with a unified framework. The domain classifier $C(\cdot)$ and discriminator of the GANs $D(\cdot)$ are first updated by minimizing Eqs. (B.1) and (3.15), respectively. The recognition model $\boldsymbol{R}(\cdot)$ and synthesis model $\boldsymbol{G}(\cdot)$ are then jointly updated by minimizing the sum of Eqs. (B.2) and (3.3). Since the loss for training $\boldsymbol{G}(\cdot)$ is also used for training $\boldsymbol{R}(\cdot)$, the recognition model can be expected to predict speaker-invariant PPGs that can accurately generate target speech parameters. Figure B.3 illustrates a schematic diagram of the proposed joint adversarial training.

B.3.2 Discussion

As shown in Figs. B.2(c) and B.2(d), the GAN-based algorithm alleviates over-smoothing of generated speech parameters. However, only using GANs cannot reduce the PPG differences as shown in Fig. B.1(b). On the other hand, the proposed algorithm using both the DAT and GANs successfully reduces the PPG differences as shown in Fig. B.1(c). This result can be expected to improve the converted speech quality.

In the GAN-based algorithm, the statistical difference between target speech parameters $\boldsymbol{y}^{(\mathrm{O})}$ and ones predicted by other speakers, i.e., $\hat{\boldsymbol{y}}^{(\mathrm{M})} = \boldsymbol{G}(\boldsymbol{R}(\boldsymbol{x}^{(\mathrm{M})}))$, can be reduced directly. This can be done by approximating the difference in the training of $D(\cdot)$ and

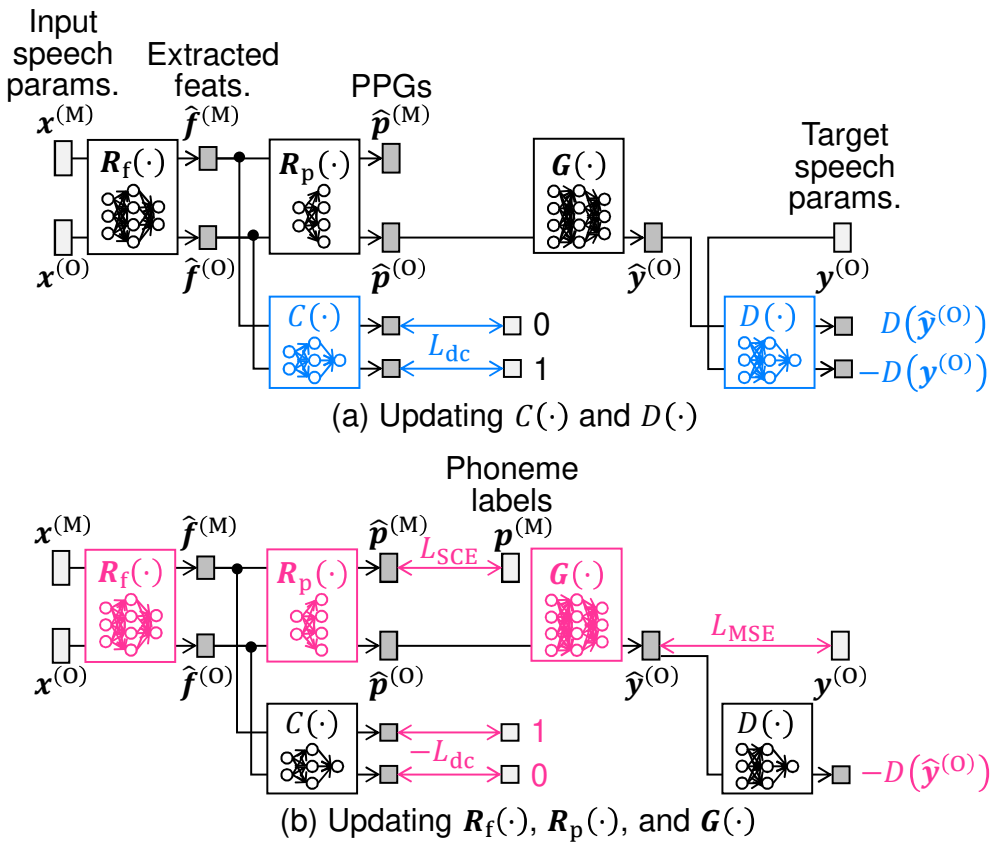

Fig. B.3. Schematic diagram of proposed joint adversarial training algorithm. Domain classifier $C(\cdot)$ and discriminator of GANs $D(\cdot)$ are first updated. Recognition model $\boldsymbol{R}(\cdot) = \boldsymbol{R}_{\mathrm{p}}(\boldsymbol{R}_{\mathrm{f}}(\cdot))$ and synthesis model $\boldsymbol{G}(\cdot)$ are then updated. These updates are iterated during training, and DNNs for many-to-one VC are constructed by concatenating final $\boldsymbol{R}(\cdot)$ and $\boldsymbol{G}(\cdot)$.

minimizing the distance in the training of $\boldsymbol{G}(\cdot)$. However, it was found from preliminary experiment results that this training considerably degraded the converted speech quality. One possible reason is that the differences among the target and other speakers in the speech parameter domain may be larger than those in the extracted feature domain, and minimizing the former using GANs becomes more difficult.

Regarding related work, several methods that incorporate GANs into the DNN training for VC have been proposed. CycleGAN-VC [108, 152] achieves non-parallel VC based on adversarial training considering cyclic-consistency [153]. StarGAN-VC [109, 154] extends this idea to many-to-many VC by introducing StarGAN [155] into the acoustic model training. Although these techniques can achieve non-parallel VC without using any text transcriptions, they cannot guarantee the quality of converted speech when the input speaker is unseen. On the other hand, the proposed algorithm can take any arbitrary source speakers for VC. Although this algorithm requires a large speech corpus with text

transcriptions and limits the target speaker to a specific one, semi-supervised training of the recognition and synthesis models (e.g., machine speech chain [115]) and conditional GAN [156] using one-hot speaker codes [41] can be expected to alleviate these limitations. Also, one can apply the proposed algorithm to more practical VC using PPGs, such as cross-lingual VC [157], one-shot VC [158], and WaveNet [159]-based VC [160, 161].

B.4 Experimental Evaluation

B.4.1 Experimental Conditions

Two professional speakers, i.e., one voice actress (FT) taken from the NICT Voice Actress Dialogue Corpus [162] and one voice actor (MT) included in an internal dataset of DeNA, were used as the target speakers for VC. The Corpus of Spontaneous Japanese (CSJ) [163] was used to train the speech recognition model $\boldsymbol{R}(\cdot)$. The CSJ included 1,417 amateur speakers (470 females and 947 males) with various speaking styles such as a monologue, dialogue, and reading aloud. About 99% of the CSJ was used as the multi-speaker corpus $\mathcal{D}^{(\mathrm{M})}$. The remainders of the CSJ was used for objective evaluations described in Section B.4.2. The FT's 5,174 utterances or MT's 2,211 utterances were used as the target speaker corpus $\mathcal{D}^{(\mathrm{O})}$ to train the speech synthesis model $\boldsymbol{G}(\cdot)$. The other 50 utterances of the target speakers were used for objective evaluations described in Section B.4.2. Note that $\mathcal{D}^{(\mathrm{M})}$ and $\mathcal{D}^{(\mathrm{O})}$ were significantly different in many aspects, such as recording environments, speaking styles, and speaking skills. All speech samples were downsampled at 16 kHz. The WORLD vocoder [48] (D4C edition [49]) was used to extract log F0, 40-dimensional mel-cepstral coefficients, and band aperiodicity. In many-to-one VC, DNNs predicted the 1st-through-39th mel-cepstral coefficients of the target speaker. The F_0 values were extracted by integrating results of multiple F_0 extractors [48, 164, 165]. The log F_0 was linearly converted. Source speaker's band aperiodicity and the 0th mel-cepstral coefficients were directly used for the speech waveform synthesis.

All DNN architectures were 1D CNNs along time axis [26] with a fixed sequence length of 128 frames. The feature extractor $\boldsymbol{R}_{\mathrm{f}}(\cdot)$ extracted 256-dimensional features from 13-dimensional MFCCs and their dynamic features. The phoneme predictor $\boldsymbol{R}_{\mathrm{p}}(\cdot)$ predicted 43-dimensional Japanese PPGs from the extracted features. The synthesis model $\boldsymbol{G}(\cdot)$ generated the 1st-through-39th mel-cepstral coefficients of the target speaker from the PPGs. The MFCCs and mel-cepstral coefficients were normalized to have zero-mean and unit-variance. The domain classifier $C(\cdot)$ distinguished $\mathcal{D}^{(\mathrm{O})}$ from $\mathcal{D}^{(\mathrm{M})}$ using the ex-

Table B.1. DNN architectures used in experimental evaluation. In this table, "Conv1D(C_{in}, C_{out} , k , s)" and "Deconv1D(C_{in}, C_{out} , k , s)" denote 1D convolution and deconvolution layers, respectively. C_{in} and C_{out} mean number of channels of input and output, respectively. k and s denote convolution window size and stride width, respectively

Recognition $\boldsymbol{R}(\cdot) = \boldsymbol{R}_{\text{p}}(\boldsymbol{R}_{\text{f}}(\cdot))$	**Synthesis $\boldsymbol{G}(\cdot)$**
Feature extractor $\boldsymbol{R}_{\text{f}}(\cdot)$	Conv1D(43, 256, 15, 1)
Conv1D(26, 256, 15, 1)	Conv1D(256, 512, 5, 2)
Conv1D(256, 512, 5, 2)	Conv1D(512, 1024, 5, 2)
Conv1D(512, 1024, 5, 2)	Deconv1D(1024, 512, 5, 2)
Deconv1D(1024, 512, 5, 2)	Deconv1D(512, 256, 5, 2)
Deconv1D(512, 256, 5, 2)	Conv1D(256, 39, 15, 1)
Phoneme predictor $\boldsymbol{R}_{\text{p}}(\cdot)$	
Conv1D(256, 43, 15, 1)	
Domain classifier $C(\cdot)$	**Discriminator of GANs $D(\cdot)$**
Conv1D(256, 512, 1, 1)	Conv1D(39, 512, 1, 1)
Conv1D(512, 512, 5, 1)	Conv1D(512, 512, 5, 1)
Conv1D(512, 512, 5, 1)	Conv1D(512, 512, 5, 1)
Conv1D(512, 1, 1, 1)	Conv1D(512, 1, 1, 1)

tracted features. The discriminator of the GANs $D(\cdot)$ distinguished natural mel-cepstral coefficients from generated ones. The activation function for hidden layers was the leaky ReLU [166]. Dropout [167] was applied to all hidden layers for avoiding overfitting. Batch normalization [168] was applied to some hidden layers in the synthesis model for accelerating the DNN training. Table B.1 shows details of the DNN architectures. In this table, "Conv1D(C_{in}, C_{out}, k, s)" and "Deconv1D(C_{in}, C_{out}, k, s)" denote 1D convolutional and deconvolutional layers, respectively. The C_{in} and C_{out} mean the number of channels of input and output, respectively. The convolution window size and stride width are denoted by k and s, respectively.

As an initial setting, the recognition model $\boldsymbol{R}(\cdot) = \boldsymbol{R}_{\text{p}}(\boldsymbol{R}_{\text{f}}(\cdot))$ was pretrained with the multi-speaker corpus $\mathcal{D}^{(\text{M})}$. The initialization was performed with one iteration using all utterances in $\mathcal{D}^{(\text{M})}$. The optimizer used for the initialization was AdaGrad [98]. The learning rate for the initialization was set to 0.01. The frame-wise phoneme prediction accuracy of the initialized recognition model was 80.4%, calculated with the evaluation data of the CSJ. Five DNNs for many-to-one VC were trained using the initialized recognition model with the following algorithms:

Baseline: Training $\boldsymbol{G}(\cdot)$ with the fixed $\boldsymbol{R}(\cdot)$ [111]

Prop. (Joint): Jointly training $\boldsymbol{R}(\cdot)$ and $\boldsymbol{G}(\cdot)$ with $\omega_{\text{C}} = 0$ and $\omega_{\text{D}} = 0$

Prop. (DAT): Jointly training $\boldsymbol{R}(\cdot)$ and $\boldsymbol{G}(\cdot)$ with $\omega_{\mathrm{C}} = 0.25$ and $\omega_{\mathrm{D}} = 0.0$
Prop. (GAN): Jointly training $\boldsymbol{R}(\cdot)$ and $\boldsymbol{G}(\cdot)$ with $\omega_{\mathrm{C}} = 0$ and $\omega_{\mathrm{D}} = 0.5$
Prop. (DAT-GAN): Jointly training $\boldsymbol{R}(\cdot)$ and $\boldsymbol{G}(\cdot)$ with $\omega_{\mathrm{C}} = 0.25$ and $\omega_{\mathrm{D}} = 0.5$

Here, the hyperparameters $(\omega_{\mathrm{C}}, \omega_{\mathrm{D}})$ were empirically chosen. All of the five algorithms were performed with five iterations using all utterances included in $\mathcal{D}^{(\mathrm{O})}$. In the training of "Prop. (*)," pairs of labeled training data $(\boldsymbol{x}^{(\mathrm{M})}, \boldsymbol{l}^{(\mathrm{M})})$ were randomly sampled from $\mathcal{D}^{(\mathrm{M})}$. The optimizers for training all DNNs, i.e., $\boldsymbol{R}(\cdot)$, $\boldsymbol{G}(\cdot)$, $C(\cdot)$, and $D(\cdot)$, were AdaGrad. The learning rates for the training were set to 0.01.

B.4.2 Objective Evaluations

Logarithmic Global Variance Distance

Since *non-parallel* many-to-one VC was considered, any objective evaluation metrics that require parallel speech utterances of source and target speakers could not be calculated. Instead, GVs [25] of the target speaker's (MT or FT) natural and generated speech were calculated. This section calculated a logarithmic GV distance (LogGVD) defined as follows:

$$\mathrm{LogGVD}\left(\boldsymbol{v}, \hat{\boldsymbol{v}}\right) = \frac{1}{M} || \log \hat{\boldsymbol{v}} - \log \boldsymbol{v} ||_2^2, \tag{B.3}$$

where $\boldsymbol{v}$ and $\hat{\boldsymbol{v}}$ denote M-dimensional GV vectors of natural and generated mel-cepstral coefficients, respectively. This evaluation would quantify the ideal performance of the many-to-one VC method because the domain mismatch in the input speech parameters never occurred. Fifty utterances of MT or FT were used to calculate the LogGVDs.

Table B.2 lists the averaged LogGVDs and their standard deviations. From the results, "Prop. (GAN)" significantly reduces the LogGVDs better than "Baseline," suggesting that the GAN-based algorithm overcomes over-smoothing of generated speech parameters in many-to-one VC. "Prop. (DAT)" also decreases the LogGVDs, and the combination of the DAT and GANs, i.e., "Prop. (DAT-GAN)," achieves the lowest value among the five algorithms. These results indicate that training the speaker-invariant recognition model increases the accuracy in modeling the target speech parameters. On the other hand, "Prop. (Joint)" shows the different tendencies in accordance with the different target speakers; i.e., its LogGVDs are almost the same as "Baseline" when the MT is used and similar to "Prop. (DAT)" in other cases. One of the reasons may be the data imbalance of the CSJ corpus that includes more male speakers than female ones.

Table B.2. LogGVDs between natural and generated mel-cepstral coefficients with their standard deviations

	Target speaker	
	MT	FT
Baseline	1.96 ± 0.34	5.05 ± 0.71
Prop. (Joint)	1.98 ± 0.38	3.93 ± 0.78
Prop. (DAT)	1.44 ± 0.26	3.80 ± 0.67
Prop. (GAN)	0.44 ± 0.18	0.21 ± 0.11
Prop. (DAT-GAN)	**0.23 ± 0.10**	**0.11 ± 0.06**

Table B.3. Frame-wise phoneme recognition accuracy of speech recognition models [%]

	Target speaker	
	MT	FT
Baseline	**80.4**	**80.4**
Prop. (Joint)	63.5	77.6
Prop. (DAT)	62.3	77.5
Prop. (GAN)	62.8	77.5
Prop. (DAT-GAN)	62.4	77.0

Speech Recognition Accuracy

Although improving the speech recognition accuracy is not the final goal of the many-to-one VC method, whether the proposed algorithms affect the accuracy or not deserves to be reported. Here, the frame-wise phoneme recognition accuracy of the recognition models after the conventional or proposed training was calculated. The evaluation data of the CSJ corpus was used to calculate the accuracy.

Table B.3 lists the evaluation results. From the results, "Baseline" achieves the highest recognition accuracy. This is a natural result since the recognition model is trained to minimize the recognition error and fixed during the synthesis model training. Meanwhile, all the proposed algorithms decrease the accuracy, suggesting that the loss functions for training the speech synthesis do not necessarily improve the speech recognition accuracy.

Speaker Invariance of Speech Recognition Model

The Matthews correlation coefficients (MCC) [169] of the domain classifier was calculated to evaluate the speech recognition model's robustness against the input speaker variation.

Table B.4. MCCs of domain classifiers

	Target speaker	
	MT	FT
Baseline	0.36	0.33
Prop. (Joint)	0.22	0.16
Prop. (DAT)	**0.02**	**0.04**
Prop. (GAN)	0.18	0.18
Prop. (DAT-GAN)	0.04	**0.04**

The MCC quantifies the performance of a binary classification model and is defined as:

$$\mathrm{MCC} = \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}}, \tag{B.4}$$

where TP, TN, FP, and FN denote the numbers of true positives, true negatives, false positives, and false negatives of the classification results, respectively. The MCC takes a value between -1 (complete misclassification) and $+1$ (perfect classification), and their mid-value (i.e., 0) corresponds to no better than random classification. Therefore, the goal in the many-to-one VC method is to cause the speech recognition model to learn features that make the MCC of the domain classifier become close to 0. Fifty utterances of the target speaker MT or FT were used as positive examples and 500 utterances (50 utterances $\times$ 10 speakers) taken from the evaluation data of the CSJ were used as negative examples to calculate the MCCs.

Table B.4 lists the evaluation results. From the results, all the proposed algorithms decrease the MCCs compared with "Baseline." In particular, the use of DAT makes the MCCs close to almost zero, while the other methods do not. These results demonstrate that the proposed algorithm using the GAN and DAT not only reduces the statistical differences between natural and generated speech parameters, but also makes the speech recognition model more invariant to the domain mismatch between the target and other speakers.

B.4.3 Subjective Evaluations

Subjective evaluations on the naturalness and speaker similarity of the converted speech were conducted. Since the speech corpora for building the many-to-one VC systems were completely non-parallel, the ATR Japanese Speech Database (set C) [170] was used for selecting source speakers. The database included 291 amateur speakers (143 females and

Table B.5. Results of MOS test on naturalness of converted speech with their 95% confidence intervals. **Bold** values indicate that method is more naturally sounded than "Baseline" with p-value < 0.05

(a) Results of FSs/MSs-to-FT VC

	FSs-to-FT	MSs-to-FT
Baseline	2.70 ± 0.12	2.51 ± 0.11
Prop. (GAN)	**3.00 ± 0.13**	2.55 ± 0.12
Prop. (DAT-GAN)	**2.95 ± 0.13**	**2.75 ± 0.12**

(b) Results of FSs/MSs-to-MT VC

	FSs-to-MT	MSs-to-MT
Baseline	2.63 ± 0.10	2.55 ± 0.11
Prop. (GAN)	**2.94** ± 0.11	**3.01 ± 0.12**
Prop. (DAT-GAN)	**2.96** ± 0.12	**2.96 ± 0.11**

148 males) with a reading-aloud style. One parallel speech utterances (phonetically balanced sentence A01) of randomly selected 10 male speakers (MSs) and 10 female speakers (FSs) were used for investigating the effects of variation in source speakers. Here, the performances of conventional and proposed algorithms were evaluated in four many-to-one VC settings: FSs-to-FT, MSs-to-FT, FSs-to-MT, and MSs-to-MT.

Evaluation of Naturalness

A five-point scaled MOS test was conducted to compare "Baseline" with the GAN-based proposed algorithms (i.e., "Prop. (GAN)" and "Prop. (DAT-GAN)") in terms of the converted speech naturalness. The converted speech generated by the three many-to-one VC systems was presented to listeners in random order. In evaluation of FSs-to-FT and MSs-to-FT VC, thirty listeners participated in the assessment by using crowdsourced subjective evaluation systems. Each listener evaluated 60 converted speech samples (20 source speakers $\times$ the three algorithms). Similarly, the evaluation of FSs-to-MT and MSs-to-MT VC was conducted with 25 listeners.

Table B.5 shows the evaluation results. "Prop. (DAT-GAN)" outperforms "Baseline" in the all VC tasks, demonstrating that the algorithm incorporating the DAT and GAN is effective in improving the naturalness of the converted speech. A noteworthy fact is that the proposed algorithm using only the GAN does not always yield the significant improvement as shown in Table B.5(a). This result suggests that just using the GAN-based training is insufficient to deal with the differences among speakers observed in PPGs.

Table B.6. Preference scores of converted speech speaker similarity. **Bold** values indicate that method is more preferred than other with p-value < 0.05. Here, "Prop. (DAT-GAN)" was compared with "Baseline" or "Prop. (GAN)"

(a) Results of FSs-to-FT VC

Method A	Score (A vs. B)	Method B
Baseline	0.317 vs. **0.683**	Prop. (DAT-GAN)
Prop. (GAN)	0.387 vs. **0.613**	Prop. (DAT-GAN)

(b) Results of MSs-to-FT VC

Method A	Score (A vs. B)	Method B
Baseline	0.283 vs. **0.717**	Prop. (DAT-GAN)
Prop. (GAN)	0.373 vs. **0.627**	Prop. (DAT-GAN)

(c) Results of FSs-to-MT VC

Method A	Score (A vs. B)	Method B
Baseline	0.328 vs. **0.672**	Prop. (DAT-GAN)
Prop. (GAN)	0.348 vs. **0.652**	Prop. (DAT-GAN)

(d) Results of MSs-to-MT VC

Method A	Score (A vs. B)	Method B
Baseline	0.308 vs. **0.692**	Prop. (DAT-GAN)
Prop. (GAN)	0.276 vs. **0.724**	Prop. (DAT-GAN)

Evaluation of Speaker Similarity

"Prop. (DAT-GAN)" was compared with "Baseline" or "Prop. (GAN)" in terms of the converted speech speaker similarity. Preference XAB tests on the similarity were conducted. The target speaker's three speech utterances excluded from the training data were used as reference "X" to evaluate the similarity. The converted speech pairs of the method "A" ("Baseline" or "Prop. (GAN)") and the method "B" ("Prop. (DAT-GAN)") were presented to listeners in random order. In the evaluation of FSs-to-FT and MSs-to-FT VC, thirty listeners participated in the assessment by using crowdsourced subjective evaluation systems. Each listener evaluated 40 converted speech samples (20 source speakers $\times$ the two comparisons). Similarly, the evaluation of FSs-to-MT and MSs-to-MT VC was conducted with 25 listeners.

Table B.6 shows the evaluation results. "Prop. (DAT-GAN)" achieves significantly higher preference scores than not only "Baseline" but also "Prop. (GAN)" in the all VC settings. These results demonstrate the algorithm's effectiveness in improving the converted speech speaker similarity, as well as in the naturalness.

Table B.7. Preference scores of converted speech naturalness. **Bold** values indicate that method is more preferred than other with p-value < 0.05. Here, "Baseline" was compared with "Prop. (Joint)" or "Prop. (DAT)"

(a) Results of FSs-to-FT VC

Method A	Score (A vs. B)	Method B
Baseline	0.468 vs. 0.532	Prop. (Joint)
Baseline	0.448 vs. **0.552**	Prop. (DAT)

(b) Results of MSs-to-FT VC

Method A	Score (A vs. B)	Method B
Baseline	0.512 vs. 0.488	Prop. (Joint)
Baseline	0.492 vs. 0.508	Prop. (DAT)

(c) Results of FSs-to-MT VC

Method A	Score (A vs. B)	Method B
Baseline	0.420 vs.**0.580**	Prop. (Joint)
Baseline	0.408 vs **0.592**	Prop. (DAT)

(d) Results of MSs-to-MT VC

Method A	Score (A vs. B)	Method B
Baseline	0.388 vs. **0.612**	Prop. (Joint)
Baseline	0.400 vs. **0.600**	Prop. (DAT)

Effects of Joint Training and DAT without GANs

The effects of the other proposed algorithms, i.e., "Prop. (Joint)" and "Prop. (DAT)," were investigated. As shown in Table B.2, these algorithms were unable to reduce Log-GVDs to the extent that GAN-based proposed algorithms were able to do. However, other objective evaluation results revealed that there were clear differences between "Baseline" and the two proposed algorithms. Therefore, this section compared "Baseline" with "Prop. (Joint)" or "Prop. (DAT)" by preference AB tests on the naturalness and preference XAB tests on the speaker similarity. The comparison should clarify the effects caused by the joint training or the DAT-based training. The converted speech pairs of the method "A" ("Baseline") and the method "B" ("Prop. (Joint)" or "Prop. (DAT)") were presented to listeners in random order. Twenty-five listeners participated in the evaluations. Each listener evaluated 40 converted speech samples (20 source speakers $\times$ the two comparisons).

Tables B.7 and B.8 show the evaluation results on the naturalness and speaker similarity, respectively. From the results, the preference scores of the two proposed algorithms are comparable or superior to those of "Baseline." In particular, two remarkable points are

Table B.8. Preference scores of converted speech speaker similarity. **Bold** values indicate that method is more preferred than other with p-value < 0.05. Here, "Baseline" was compared with "Prop. (Joint)" or "Prop. (DAT)"

(a) Results of FSs-to-FT VC

Method A	Score (A vs. B)	Method B
Baseline	0.500 vs. 0.500	Prop. (Joint)
Baseline	0.484 vs. 0.516	Prop. (DAT)

(b) Results of MSs-to-FT VC

Method A	Score (A vs. B)	Method B
Baseline	0.456 vs. 0.544	Prop. (Joint)
Baseline	0.432 vs. **0.568**	Prop. (DAT)

(c) Results of FSs-to-MT VC

Method A	Score (A vs. B)	Method B
Baseline	0.424 vs. **0.576**	Prop. (Joint)
Baseline	0.404 vs. **0.596**	Prop. (DAT)

(d) Results of MSs-to-MT VC

Method A	Score (A vs. B)	Method B
Baseline	0.408 vs. **0.592**	Prop. (Joint)
Baseline	0.396 vs. **0.604**	Prop. (DAT)

observed: 1) "Prop. (DAT)" improves the speaker similarity in the inter-gender VC settings (i.e., MSs-to-FT and FSs-to-MT) and 2) the two proposed algorithms outperform "Baseline" in FSs-/MSs-to-MT VC regarding both the naturalness and speaker similarity in spite of decreasing the speech recognition accuracy shown in Table B.3. These results suggest that 1) without using the GANs, the DAT has the effect of improving the speaker similarity by learning speaker-invariant features in the recognition model, and 2) the high speech recognition accuracy does not guarantee the quality of converted speech and the proposed joint training can optimize the recognition model for the many-to-one VC.

Comparison with Another Non-parallel VC Method

The best proposed algorithm, i.e., "Prop. (DAT-GAN)," was compared with another state-of-the-art non-parallel VC method. Here, StarGAN-VC [109] was used as the competitive VC method that can achieve high-quality many-to-many VC without requiring parallel speech corpora. The StarGAN-VC model was trained using an open-source imple-

Table B.9. Preference scores of converted speech naturalness. **Bold** values indicate that method is more preferred than other with p-value < 0.05. Here, "StarGAN-VC" was compared with "Prop. (DAT-GAN)"

(a) Results of FSs-to-FT VC

Method A	Score (A vs. B)	Method B
StarGAN-VC	0.168 vs. **0.832**	Prop. (DAT-GAN)

(b) Results of MSs-to-FT VC

Method A	Score (A vs. B)	Method B
StarGAN-VC	0.152 vs. **0.848**	Prop. (DAT-GAN)

(c) Results of FSs-to-MT VC

Method A	Score (A vs. B)	Method B
StarGAN-VC	0.300 vs. **0.700**	Prop. (DAT-GAN)

(d) Results of MSs-to-MT VC

Method A	Score (A vs. B)	Method B
StarGAN-VC	0.400 vs. **0.600**	Prop. (DAT-GAN)

mentation[*1] and 100 utterances of 22 speakers (MSs, FSs, MT, and FT) for the training. Similar to Section B.4.3, preference (X)AB tests were conducted with 25 listeners.

Tables B.9 and B.10 show the evaluation results on the naturalness and speaker similarity, respectively. The results demonstrate that "Prop. (DAT-GAN)" outperforms not only the conventional many-to-one VC training algorithm but also state-of-the-art non-parallel VC using the StarGAN.

[*1] `https://github.com/hujinsen/pytorch-StarGAN-VC`

Table B.10. Preference scores of converted speech speaker similarity. **Bold** values indicate that method is more preferred than other with p-value < 0.05. Here, "StarGAN-VC" was compared with "Prop. (DAT-GAN)"

(a) Results of FSs-to-FT VC

Method A	Score (A vs. B)	Method B
StarGAN-VC	0.140 vs. **0.860**	Prop. (DAT-GAN)

(b) Results of MSs-to-FT VC

Method A	Score (A vs. B)	Method B
StarGAN-VC	0.096 vs. **0.904**	Prop. (DAT-GAN)

(c) Results of FSs-to-MT VC

Method A	Score (A vs. B)	Method B
StarGAN-VC	0.132 vs. **0.868**	Prop. (DAT-GAN)

(d) Results of MSs-to-MT VC

Method A	Score (A vs. B)	Method B
StarGAN-VC	0.176 vs. **0.824**	Prop. (DAT-GAN)

Appendix C

Speech Recognition and Speaker Classification Performances

This appendix describes the speech recognition and speaker classification performances in VAE-based multi-speaker statistical speech synthesis (Chapter 5). Several settings of two hyperparameters, the number of seen speakers and d-vector dimensionality, were considered. Two performances were measured: the frame-wise phoneme error rate (PER) of speech recognition and equal error rate (EER) of speaker verification using d-vectors. For measuring the EER, 50 utterances of each of the six speakers (three males and three females) were used for enrollment (i.e., d-vector extraction) and other 25 utterances were used for evaluation (i.e., computing the cosine distance between input and claimed speaker's d-vectors). The L2 normalization was applied to the d-vectors [27].

Table C.1 shows the results. The PER significantly increases under the setting of the smallest number of seen speakers (i.e., "50spk"). The results also reveal that the EER drastically increases under the settings of the smaller d-vector dimensionality (i.e., "1d," "2d," and "4d").

Table C.1. Frame-wise PER of speech recognition DNNs and EER of speaker verification using d-vectors. Different settings of number of seen speakers and d-vector dimensionality were considered. Note that, EER for one-dimensional *d*-vectors cannot be computed since *d*-vector in this evaluation always takes positive value and is normalized so that its L2 norm becomes 1

	PER [%]	EER [%]					
		1d	2d	4d	8d	16d	32d
50spk	54.2	N/A	29.5	10.4	6.96	2.97	2.48
130spk	48.6	N/A	29.1	13.1	7.81	1.60	1.33
260spk	49.2	N/A	27.4	10.3	6.22	4.07	0.78