Scaling and Bias Codes for Modeling Speaker-Adaptive DNN-based Speech Synthesis Systems

SCALING AND BIAS CODES FOR MODELING SPEAKER-ADAPTIVE DNN-BASED SPEECH SYNTHESIS SYSTEMS

Hieu-Thi Luong[1], Junichi Yamagishi[1,2]

[1]National Institute of Informatics, Tokyo, Japan [2]The University of Edinburgh, Edinburgh, UK

ABSTRACT

Most neural-network based speaker-adaptive acoustic models for speech synthesis can be categorized into either layer-based or input-code approaches. Although both approaches have their own pros and cons, most existing works on speaker adaptation focus on improving one or the other. In this paper, after we first systematically overview the common principles of neural-network based speaker-adaptive models, we show that these approaches can be represented in a unified framework and can be generalized further. More specifically, we introduce the use of scaling and bias codes as generalized means for speaker-adaptive transformation. By utilizing these codes, we can create a more efficient factorized speaker-adaptive model and capture advantages of both approaches while reducing their disadvantages. The experiments show that the proposed method can improve the performance of speaker adaptation compared with speaker adaptation based on the conventional input code.

Index Terms— speech synthesis, speaker adaptation, neural network, factorization, speaker code

1. INTRODUCTION

Recent speaker-dependent speech synthesis systems can generate high-quality reading speech indistinguishable from natural human speech when their training data is recorded in a quality-controlled condition and have sufficient amount of data [1]. The speech synthesis community is currently trying to solve more challenging problems. A good example is multi-speaker speech synthesis and its adaptation [2, 3, 4, 5]. Here multi-speaker synthesis means generating synthetic speech of multiple known speakers included in a training dataset using a common model, and adaptation means adapting the speaker-independent common model to unseen speakers and generating their speech. This speaker-adaptive speech synthesis systems are expected to opens possibilities for a wide range of new applications for speech synthesis such as a customizable, user-specific voice interface and voice preservation for people with medical conditions involving voice losses. However, training the multi-speaker synthesis models and adapting them to unseen speakers are still challenging problems, and resulting models are far from perfect, especially when less than ideal datasets are used [6].

Most adaptation methods for neural network models can be described as either (a) fine-tuning a set of or all of parameters of speaker-independent network so it explains unseen speaker's data better or (b) factorizing a neural network into speaker-specific and common parts and estimating the speaker-specific components for the unseen speaker's data. The speaker-specific components may be composed by input codes (e.g. one-hot vector) [7], embedding vectors obtained externally (e.g. i-vector) [8], or latent variables (e.g. variational auto-encoder) [3, 9, 10]. Of course any of those speaker-specific components may be jointly optimized with the common parts (e.g. [7, 10, 11]). Although there are a lot of variants on multi-speaker modeling and adaptation, most approaches for augmenting the speaker-specific components into a neural network are equivalent to adapting a bias term of each hidden layer and this bias term is typically constant across all frames of all utterances. Although Wu et al. [12] and Nachmachi et al. [13] proposed frame-dependent components, these components are still bias adaptation and their underlying frameworks and concepts have mathematical similarities.

In this paper we first systematically overview the common concepts of neural-network based speaker-adaptive models and show that these approaches can be represented in a unified framework. Further, we introduce a scaling code as an extended speaker-adaptive transformation. As its name indicates, this code introduces an additional scaling operation as an approximation to adaptation of weight matrices unlike the conventional deep neural network (DNN) adaptation approaches. Section 2 details relevant work. Section 3 describes our factorized speaker adaptation based on scaling and bias codes. Section 4 explains our experiments and shows both objective and subjective results. We conclude our work and describe the future direction for this method in Section 5.

2. RELATED WORK

Constrained Maximum Likelihood Linear Regression (CMLLR) [14, 15], also known as feature-space MLLR (fMLLR), is a widely used speaker adaptation technique for hidden

This work was partially supported by MEXT KAKENHI Grants (16H06302, 17H04687, 18H04120, and 18H04112).

Markov model (HMM)-based speech processing systems in which a speaker-dependent affine transformation is applied to source acoustic features to explain target data better. In the case of automatic speech recognition (ASR), the transformation acts as a method of normalization, whereas in the case of speech synthesis, the transformation purpose is to diverge the acoustic output to each target speaker [16]. The fMLLR method can be described using the following equation:

$$\overline{\mathbf{x}} = \mathbf{A}^{(k)}\mathbf{x} + \mathbf{b}^{(k)} \tag{1}$$

where $\mathbf{x}$ is the source acoustic features, $\overline{\mathbf{x}}$ represents approximated acoustic features of the target speaker k, $\mathbf{A}^{(k)}$ is a full linear matrix and $\mathbf{b}^{(k)}$ is the bias vector. $\mathbf{A}^{(k)}$ and $\mathbf{b}^{(k)}$ are transformation parameters specific to each speaker.

A feedforward layer of a standard neural network can be defined by the following equation:

$$\mathbf{h}_l = f(\mathbf{W}_l\mathbf{h}_{l-1} + \mathbf{c}_l) \tag{2}$$

where $\mathbf{h}_l$ is the output of the l-th hidden layer. To simplify our equation, let us assume all hidden layers have the same number of hidden units m, that is, $\mathbf{h}_l, \mathbf{h}_{l-1} \in \mathbb{R}^{m\times 1}$ and the l-th hidden layer has a weight matrix $\mathbf{W}_l \in \mathbb{R}^{m\times m}$ and a bias vector $\mathbf{c}_l \in \mathbb{R}^{m\times 1}$. $f(.)$ is an element-wise non-linear activation function (such as sigmoid or tanh) that deterministically squashes each dimension of an input vector $\mathbb{R}^{m\times 1}$ to a limited range.

Next we explain the existing DNN-based speaker adaptation methods, that is, speaker-dependent layers and speaker-dependent input code using similar notations to the above fMLLR. For the speaker-dependent layers [17, 18] approach, the weight matrices and bias vectors of specific layers are fine-tuned using adaptation data, therefore we can rewrite Equation 2 as:

$$\overline{\mathbf{h}}_l = f(\mathbf{W}_l^{(k)}\mathbf{h}_{l-1} + \mathbf{c}_l^{(k)}) \tag{3}$$

where $\mathbf{W}_l^{(k)}$ and $\mathbf{c}_l^{(k)}$ are now specific to a target speaker k and $\overline{\mathbf{h}}_l$ also represents an adapted hidden layer . The method has the advantage of modeling both a full matrix $\mathbf{W}_l^{(k)}$ and the bias vector $\mathbf{c}_l^{(k)}$, which usually yield favorable result when the adaptation data is sufficient [8, 18]. However when the amount of adaptation data is limited, the result is unstable as number of parameters estimated is very large [19]. This is also the reason that this method typically involves reducing the number of parameters estimated [20, 21, 18] in order to retain the adaptation performance.

Learning Hidden Unit Contribution (LHUC) [22] is an adaptation method that transforms outputs of the activation function using a speaker-dependent diagonal transformation matrix, which significantly reduces the number of parameters:

$$\overline{\mathbf{h}}_l = \mathrm{Diag}\mathbf{A}_l^{(k)} \circ f(\mathbf{W}_l\mathbf{h}_{l-1} + \mathbf{c}_l) \tag{4}$$

where $\mathbf{A}_l^{(k)} \in \mathbb{R}^{m\times m}$ is a diagonal matrix for speaker k, Diag is an operation to extract diagonal elements of a $m \times m$ matrix as a $m \times 1$ vector, and $\circ$ is an element-wise multiplication of vectors. In LHUC, since we apply the transformation after the activation function of the current layer, we may write the LHUC operation at the next hidden layer as follows:

$$\begin{aligned}
\overline{\mathbf{h}}_{l+1} &= f(\mathbf{W}_{l+1}\overline{\mathbf{h}}_l + \mathbf{c}_{l+1}) && (5)\\
&= f\left(\mathbf{W}_{l+1}\cdot\left(\mathrm{Diag}\mathbf{A}_l^{(k)} \circ \mathbf{h}_l\right) + \mathbf{c}_{l+1}\right) && (6)\\
&= f\left(\mathbf{W}_{l+1}\mathbf{A}_l^{(k)}\mathbf{h}_l + \mathbf{c}_{l+1}\right) && (7)\\
&= f\left(\mathbf{W}_{l+1}^{(k)}\mathbf{h}_l + \mathbf{c}_{l+1}\right) && (8)
\end{aligned}$$

From these equations, we see that a speaker-specific weight matrix $\mathbf{W}_{l+1}^{(k)}$ is factorized as $\mathbf{W}_{l+1}\mathbf{A}_l^{(k)}$.

For the speaker-dependent input-code approach, a vector representing the speaker identity is fed into one or many layers of a neural network. This vector can be as simple as an one-hot vector [7, 19] or an embedding vector obtained from outside systems like speaker verification [6, 23] or speaker recognition [24]. Although there are many variations, each may be viewed as a bias adaptation of a hidden layer and the speaker-dependent input approach can be written as:

$$\begin{aligned}
\overline{\mathbf{h}}_l &= f(\mathbf{W}_l\mathbf{h}_{l-1} + \mathbf{c}_l + \mathbf{W}_l^b\mathbf{s}^{(k)}) && (9)\\
&= f(\mathbf{W}_l\mathbf{h}_{l-1} + \mathbf{c}_l^{(k)}) && (10)
\end{aligned}$$

where $\mathbf{s}^{(k)} \in \mathbb{R}^{q\times 1}$ is the auxiliary input vector specific to speaker k and has an arbitrary size q; $\mathbf{W}_l^b \in \mathbb{R}^{m\times q}$ is a new weight matrix added to the layer to handle the new input. The input code approach provides the flexibility of using an outside system to constrain the model. It is also convenient to present each speaker (or speaking style) as one single vector since it may be used for controlling characteristics of synthetic speech [7, 3, 25]. As the number of speaker-dependent parameters q is typically small, this method shows preferable results when the amount of adaptation data is limited. However, it does not seem to improve the adaptation performance when the adaptation data is plentiful [19].

3. FACTORIZED SPEAKER TRANSFORMATION BASED ON SCALING AND BIAS CODES

3.1. Scaling and bias codes

The above approaches are obviously complementary. Our proposal, illustrated in Figure 1, is therefore the design of a new speaker transformation by combining the above two types of approaches and further factorizing its essential components on the basic of "scaling" and "bias" codes. The main idea is to explicitly transform both the weight matrix and the bias vector as:

$$\begin{aligned}
\overline{\mathbf{h}_l} &= f(\mathbf{A}_l^{(k)}\mathbf{W}_l\mathbf{h}_{l-1} + \mathbf{c}_l + \mathbf{b}_l^{(k)}) && (11)\\
\mathbf{A}_l^{(k)} &= \mathrm{diag}(\mathbf{W}_l^A\mathbf{s}^{A,(k)}) && (12)\\
\mathbf{b}_l^{(k)} &= \mathbf{W}_l^b\mathbf{s}^{b,(k)} && (13)
\end{aligned}$$

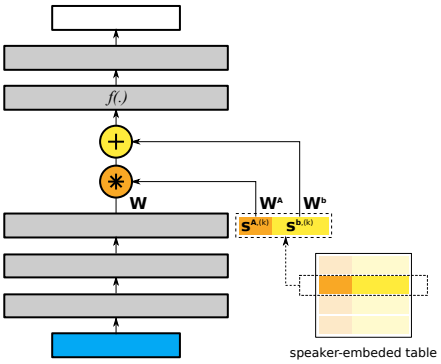


Fig. 1. Proposed factorized speaker transformation based on scaling and bias codes. Gray boxes indicate layers with non-linear activation function, and the white box indicates a layer with linear function.

where $\mathbf{A}_l^{(k)} \in \mathbb{R}^{m\times m}$ is a diagonal matrix for the scaling operation at the l-th layer. The matrix is further factorized into a speaker-independent projection matrix $\mathbf{W}_l^A \in \mathbb{R}^{m\times p}$ and a scaling code vector $\mathbf{s}^{A,(k)} \in \mathbb{R}^{p\times 1}$. diag is an operation to change a $m \times 1$ vector into a diagonal $m \times m$ matrix. The speaker-specific bias term $\mathbf{b}_l^{(k)}$ is also factorized in the same way using $\mathbf{W}_l^b \in \mathbb{R}^{m\times q}$ and $\mathbf{s}^{b,(k)} \in \mathbb{R}^{q\times 1}$. As described previously, $\mathbf{s}^{b,(k)}$ is basically equivalent to the conventional speaker code, but we call it as bias code here to better outline its property. These codes may have arbitrary lengths, but, p and q are usually chosen to be much smaller than m to reduce the number of free parameters further.

Factorizing models explicitly and using lower-dimensional subspaces is a powerful concept used in various models (e.g. Heteroscedastic Linear Discriminant Analysis (HLDA) [26], subspace Gaussian mixture model [27]). The proposed factorization is somewhat similar to Factorize Hidden Layer (FHL) introduced by Samrakoon and Sim [20], but we focus on performing the scaling and bias adaptation simultaneously using lower dimensional vectors. A concept similar to scaling and bias codes was also investigated for ASR in [28, 29], but instead of mapping the scaling and bias transformation from a common vector we use separated vectors as scaling and bias codes to give ourselves more degrees of freedom to design a speaker-adaptive architecture. If necessary, we may directly adapt $\mathbf{A}_l^{(k)}$ and $\mathbf{b}_l^{(k)}$ when the amount of adaptation data is sufficient.

3.2. Extensions of the proposed method

In this paper, we investigate two more strategies as extensions of the proposed method. The first strategy is to separately use the scaling and bias codes at different layers and to explicitly perform either scaling or bias operations only as illustrated by Figure 2-a. This is a special case of the proposed method.

The second strategy is to combine the proposed method with other type of matrix decomposition. For example, in the work of Xue et al. [30], a weight matrix is decomposed into three linearly connected matrices using singular value decomposition (SVD). Therefore, instead of multiplying a scaling matrix to a weight matrix, we may first decompose the weight matrix into the three linearly connected matrices and use the proposed scaling matrix to approximate one of the decomposed matrices further as follows:

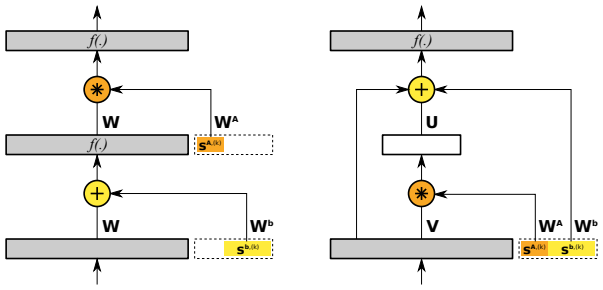
$$\bar{\mathbf{h}}_l = f(\mathbf{W}_l^{(k)}\mathbf{h}_{l-1} + \mathbf{c}_l + \mathbf{b}_l^{(k)} + \mathbf{h}_{l-1}) \quad (14)$$

$$\mathbf{W}_l^{(k)} = \mathbf{U}_l\mathbf{A}_l^{(k)}\mathbf{V}_l \quad (15)$$

$$\mathbf{A}_l^{(k)} = \mathrm{diag}(\mathbf{W}_l^A\mathbf{s}^{A,(k)}) \quad (16)$$

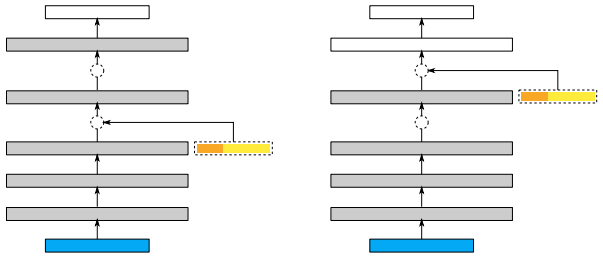
$$\mathbf{b}_l^{(k)} = \mathbf{W}_l^b\mathbf{s}^{b,(k)} \quad (17)$$

where $\mathbf{U}_l \in \mathbb{R}^{m\times n}$, $\mathbf{V}_l \in \mathbb{R}^{n\times m}$ and $\mathbf{A}_l^{(k)} \in \mathbb{R}^{n\times n}$ with $n \ll m$[1]. Note that residual connections are also added here. When we use this model for time-series speech data, the input varies at each time and the residual part becomes a time-variant bias term as $\bar{\mathbf{h}}_{l,t} = f(\mathbf{W}_l^{(k)}\mathbf{h}_{l-1,t}+\mathbf{c}_l+\mathbf{b}_l^{(k)}+\mathbf{h}_{l-1,t})$ where $\mathbf{h}_{l,t}$ is output of the l-th hidden unit at time t. The bottleneck method can be summarized as Figure 2-b.



(a) Multilevel (b) Bottleneck

Fig. 2. Extended strategies utilizing the scaling and bias codes to integrate speaker transformations into neural network



(a) Nonlinear case (b) Linear case

Fig. 3. Different injection points of proposed factorized speaker transformation. It may be applied to intermediate hidden layers with non-linear activation functions or used at a specific layer where all remaining operations are linear. Relationships between speaker transforms and acoustic features are non-linear for the former case but linear for the latter case.

[1]It is also possible to theoretically include SVD bottleneck speaker adaptation with low-rank approximation [31]. To do this, a constrain $\mathbf{W}_l \approx \mathbf{U}_l\mathbf{V}_l$ needs to be added.

Table 1. Divisions of English and Japanese speech corpora used in our experiments.

Set	Train (Speech & Text)		Valid (Speech & Text)		Test (Text)		Speakers		
	Each speaker	Total	Each speaker	Total	Each speaker	Total	Male	Female	Total
en.base	∼370	26785	5	360	-	-	31	41	72
en.target.10	10	80	5	40	15	120	4	4	8
en.target.40	40	500							
en.target.160	160	1280							
en.target.320	320	2560							
jp.base	∼148	34713	3	705	-	-	51	184	235
jp.target.10	10	200	3	60	10	200	10	10	20
jp.target.50	50	1000							
jp.target.100	100	2000							

Table 2. Different strategies evaluated in this paper. The parameter's size was purposely chosen so that all models used the same number of parameters.

Notation	Strategy	Size		
		Scaling	Bias	Bottleneck
bias	bias code	-	64	-
scale	scaling code	64	-	-
affine	bias + scaling	32	32	-
level	multilevel	32	32	-
bottle	bottleneck	64	32	512

We also investigate to which layers we should inject the proposed transformation and what kinds of activation functions should be used after the speaker transformation. More specifically, we investigate whether the proposed transformation should be used at intermediate hidden layers with non-linear activation functions as shown in Figure 3-a or at a specific layer where all remaining operations are linear as shown in Figure 3-b. By analyzing this, we can understand whether the relationship between the proposed speaker transformation functions and generated acoustic features should be represented in a non-linear way like the former case, or in a linear one like the latter case.[2]

4. EXPERIMENTS

4.1. Experimental condition

We use two speech corpora to evaluate our proposal: an English corpus containing 80 speakers, which is a subset of the VCTK [32, 33], and an in-house Japanese speech corpus with over 250 speakers. The English corpus was used to objectively evaluate various aspects of our proposal while the Japanese corpus is used to reproduce the results and evaluate subjectively with native Japanese listeners. We split each corpora into the base and target sets as shown in Table 1 and conducted two tasks (multi-speaker and adaptation)

[2]For the combination of the linear case with the strategy in Figure 2-a, which has operations at two different layers, we first used speaker transformation based on the bias code at a hidden layer with the non-linear activation functions and further used speaker transformation based on the scaling code at the next linear layer. This is technically a mix of linear and non-linear speaker transformations, but we included this in "the linear setup" in our experiments.

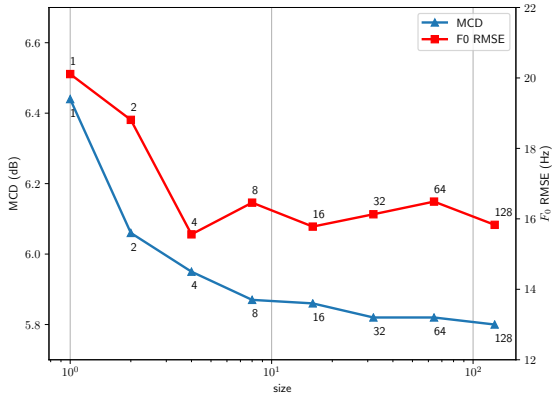


Fig. 4. Objective evaluation of changing size of scaling code in nonlinear setup.

as follows. In the multi-speaker task, we used en.base and one of en.target.{10, 40, 160, or 320} for training a multi-speaker neural network common to all speakers per strategy. In the adaptation task, we used en.base for training a multi-speaker neural network per strategy and adapted it to each target speaker included in en.target.*. In both the tasks, the evaluation was performed using target speakers included in en.target.*. This increased the number of models needed to be constructed but reduced the mismatch between the multi-speaker and adaptation tasks so we could directly compare them.

For the DNN-based acoustic model, we used a conventional multi-task learning neural network similar to our previous works [7, 34]. The neural network maps linguistic features (depending on languages) to several acoustic features including 60-dimensional mel-cepstral coefficients, 25-dimensional band-limited aperiodicities, interpolated logarithm fundamental frequencies, and their dynamic counterpart. A voiced/unvoiced binary flag is also included. The neural network model has five feedforward layers each with 1024 neurons, followed by a linear layer to map to the desired dimensional output. All layers have the sigmoid activation function unless stated otherwise. We experimented with five strategies utilizing either scaling code, bias code, or both as shown in Table 2. Further, to investigate the impacts of different waveform generation methods, we used both a speaker-independent Wavenet vocoder [35, 36] and the WORLD vocoder [37] for speech waveform generation .

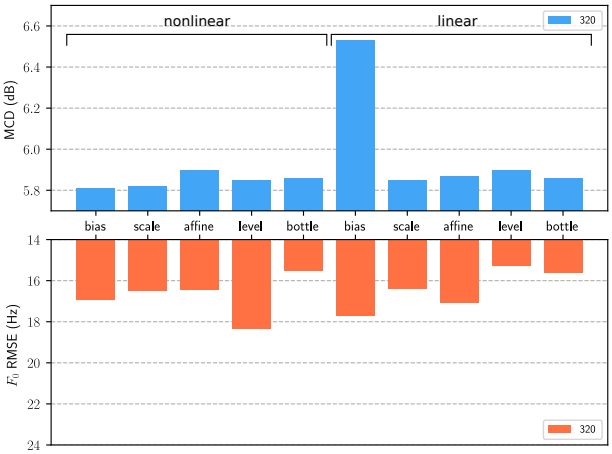


Fig. 5. Objective evaluations results of different strategies in the multi-speaker task using the English corpus.

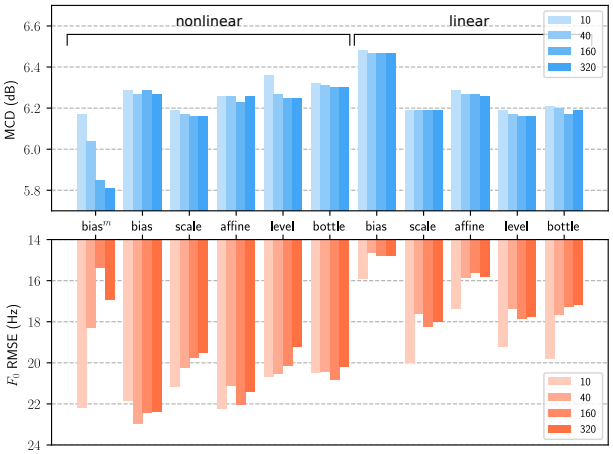


Fig. 6. Objective evaluation results of different strategies in adaptation task using English corpus. Here bias^m shows reference results in the multi-speaker task using the bias code in the nonlinear setup. All other results are for adaptation of unseen speakers using data included in en.target.*.

However, our Wavenet model is still under development and we experienced the collapse of generated speech problems, which is described in [38].

4.2. Objective evaluation

We first evaluated the scaling code by itself in a nonlinear setup since, at the time of writing, using scaling code for multi-speaker speech synthesis has not been investigated. We changed the size of scaling codes from 1 to 128 to see how they impact the objective performance of the multi-speaker task in a similar way to experiments that we did on bias codes previously [7]. The multi-speaker models were trained using en.base and en.target.320 together. The objective evaluation results, including mel-cepstral distortion (MCD) in dB and F_0 root mean square error (F_0 RMSE) in Hz, are illustrated in Figure 4. We can see that both the distortions decrease when we increase the size of the scaling code.

Next we evaluated multiple strategies described in Table 2 for the multi-speaker task in either nonlinear or linear setups. Again the multi-speaker models were trained using the en.base and en.target.320 data together. Figure 5 shows objective evaluation results of the strategies. If we look at the non-linear setups, we see that there are no obvious differences between these strategies. However, at least we can determine that the proposed scaling code can be used by itself without decreasing the performance. If we look at the linear setups, we can clearly see that the using the bias code by itself is a poor strategy for multi-speaker modeling. It resulted in much worse MCD even though its F_0 RMSE is comparable to other systems. In [39], Wang found out that the model structures required for mel-cepstrum and fundamental frequency are different. Our results also support this finding.

Figure 6 shows objective evaluation results of the strategies in the adaptation task using different amounts of data. The first block indicated bias^m corresponds to reference results in the multi-speaker task (i.e., systems where multi-speaker neural networks were trained using en.base and one of en.target.{10, 40, 160, or 320} and synthetic speech was generated using text of the test set of target speakers) using the bias code in the nonlinear setup. All other results are adaptation results for the unseen speaker task. The amounts of adaptation data vary from 10 to 320.

From this figure, we see that adaptation to the unseen speakers is more difficult than multi-speaker modeling. Moreover, while the results of multi-speaker modeling are improved significantly when we increase the amount of data, the adaptation results for the unseen speakers show marginal improvements when more data is available. This suggests that the proposed adaptation transformation needs to be generalized better. Another important pattern that we can see from the figure is that in terms of F_0 RMSE, all strategies in the linear setup outperform their nonlinear counterparts.

4.3. Subjective evaluations

Next we reproduced several selected strategies using the Japanese dataset. We doubled the size of speaker codes shown in Table 2 and chose strategies that showed reasonable improvements in the objective evaluation using the English dataset. The objective evaluation results using the Japanese corpus are shown in Figure 7, from which we can see the same trend as the result using the English one[3].

We used the Japanese systems and conducted a subjective listening test to see how participants perceived these differences. The listening test contained two sets of questions. In the first part, participants were asked to judge the naturalness of the presented speech sample using a five-point scale

[3]Speech samples using the English corpus can be found at `http://www.hieuthi.com/papers/slt2018`

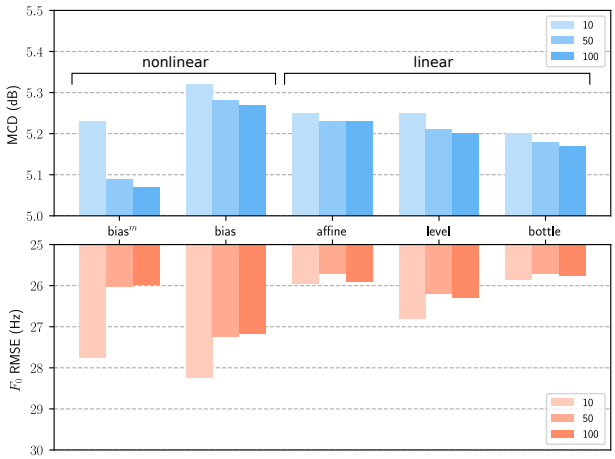


Fig. 7. Objective evaluation results of selected strategies in adaptation task using Japanese corpus. Like the English test, bias^m shows reference results in the multi-speaker task using the bias code in the nonlinear setup. All other results are adaptation results.

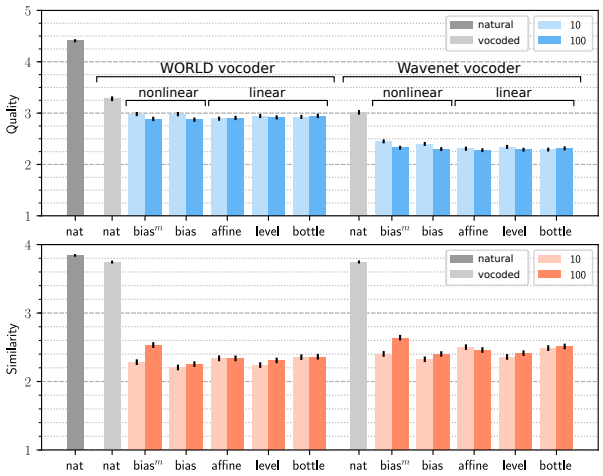


Fig. 8. Subjective evaluation results of selected strategies in adaptation task using Japanese corpus. Top figure shows mean opinion scores on naturalness. Bottom figure shows speaker similarity scores. Recorded speech and vocoded speech using correct acoustic features were also evaluated at the same time.

ranged from 1 (very unnatural) to 5 (very natural). In the second part, participants were asked to compare a speech sample of a system with recorded speech of the same speaker and judge if they are the same speaker or not using a four-point scale ranged from 1 (different, sure) to 4 (same, sure). This evaluation methodology is similar to our previous study [34]. In addition to synthetic speech generated from the proposed speech synthesis systems using the above selected strategies, we also evaluated recorded speech, WOLRD vocoded speech, and Wavenet vocoded speech for comparison. A large-scale listening test was done with 289 subjects. The statistical analysis was conducted using pairwise t-tests with a 95% confidence margin and Holm-Bonferroni compensation for multiple comparisons.

Subjective evaluation results are presented in Figure 8. In the quality test, we can first see that participants judged all systems using our speaker-independent Wavenet vocoder samples to be worse than counterparts using the WORLD vocoder. This is inconsistent with other publication results and indicates that our Wavenet is not properly trained. For the future works, we could further fine-tune a part of the speaker-independent Wavenet model to stabilize the neural-net vocoder [40, 41]. However, unlike the quality test, the subjects judged synthetic speech using the Wavenet vocoder to be closer to the target speakers in the speaker similarity test although there are still large gaps between vocoded speech and synthetic speech.

We can also see that a reference multi-speaker system marked as bias^m using 100 utterances has the highest similarity score among the other systems, and this is consistent with the objective evaluation results. Regarding the adaptation to the unseen speakers, we could see that the proposed method using both the scaling and bias codes and its bottleneck variant (in the linear setting) have better results than the adaptation method using the bias code in the nonlinear setting (which is our previous work) for both WORLD and Wavenet vocoders. This would be because of improved F0 adaptation, as we can see objectively in Figure 7. Regarding the quantity of the adaptation data, more data seems to slightly improve speaker similarity of synthetic speech in general but does not improve the perception of quality. In some cases, it makes the quality of synthetic speech slightly worse.

5. CONCLUSIONS

In this paper, we have explained several major existing adaptation frameworks for DNN speech synthesis and showed one generalized speaker-adaptive transformation. Further, we have factorized the proposed transformation on the basic of scaling and bias codes and investigated its variants such as bottleneck.

From objective and subjective experiments, we showed that the proposed method, specifically the ones using both the scaling and bias codes in the linear setting, can reduce acoustic errors and improve subjective speaker similarity in the adaptation of unseen speakers . Moreover, our results clearly indicate that there are still large gaps between vocoded speech and synthetic speech in terms of speaker similarity and this clearly indicates that there is room for improving multi-speaker modeling and speaker adaptation.

Our future work includes comparing our method with other adaptation methods such as LHUC and SVD bottleneck speaker adaptation with low-rank approximation. Another interesting experiment we would like to see is the use of i-vector or d-vector [24] as a scaling code.

6. REFERENCES

[1] Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, RJ Skerry-Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis, and Yonghui Wu, "Natural TTS synthesis by conditioning WaveNet on Mel spectrogram predictions," in *Proc. ICASSP*, 2018, pp. 4779–4783.

[2] Jonathan Parker, Yannis Stylianou, and Roberto Cipolla, "Adaptation of an expressive single speaker deep neural network speech synthesis system," in *Proc. ICASSP*, 2018, pp. 5309–5313.

[3] Yuxuan Wang, Daisy Stanton, Yu Zhang, RJ Skerry-Ryan, Eric Battenberg, Joel Shor, Ying Xiao, Fei Ren, Ye Jia, and Rif A Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," *arXiv preprint arXiv:1803.09017*, 2018.

[4] Jaime Lorenzo-Trueba, Gustav Eje Henter, Shinji Takaki, Junichi Yamagishi, Yosuke Morino, and Yuta Ochiai, "Investigating different representations for modeling and controlling multiple emotions in DNN-based speech synthesis," *Speech Commun.*, vol. 99, pp. 135–143, 2018.

[5] Katsuki Inoue, Sunao Hara, Masanobu Abe, Nobukatsu Hojo, and Yusuke Ijima, "An investigation to transplant emotional expressions in DNN-based TTS synthesis," in *Proc. APSIPA ASC*, 2017, pp. 1253–1258.

[6] Ye Jia, Yu Zhang, Ron J Weiss, Quan Wang, Jonathan Shen, Fei Ren, Zhifeng Chen, Patrick Nguyen, Ruoming Pang, Ignacio Lopez Moreno, et al., "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," *arXiv preprint arXiv:1806.04558*, 2018.

[7] Hieu-Thi Luong, Shinji Takaki, Gustav Eje Henter, and Junichi Yamagishi, "Adapting and controlling DNN-based speech synthesis using input codes," in *Proc. ICASSP*, 2017, pp. 4905–4909.

[8] Zhizheng Wu, Pawel Swietojanski, Christophe Veaux, Steve Renals, and Simon King, "A study of speaker adaptation for DNN-based speech synthesis," in *Proc. Interspeech*, 2015, pp. 879–883.

[9] Kei Akuzawa, Yusuke Iwasawa, and Yutaka Matsuo, "Expressive speech synthesis via modeling expressions with variational autoencoder," *arXiv preprint arXiv:1804.02135*, 2018.

[10] Marc Delcroix, Shinji Watanabe, Atsunori Ogawa, Shigeki Karita, and Tomohiro Nakatani, "Auxiliary feature based adaptation of end-to-end asr systems," in *Proc. Interspeech*, 2018, pp. 2444–2448.

[11] Moquan Wan, Gilles Degottex, and Mark JF Gales, "Integrated speaker-adaptive speech synthesis," in *Proc. ASRU*, 2017, pp. 705–711.

[12] Xixin Wu, Lifa Sun, Shiyin Kang, Songxiang Liu, Zhiyong Wu, Xunying Liu, and Helen Meng, "Feature based adaptation for speaking style synthesis," in *Proc. ICASSP*, 2018, pp. 5304–5308.

[13] Eliya Nachmani, Adam Polyak, Yaniv Taigman, and Lior Wolf, "Fitting new speakers based on a short untranscribed sample," *arXiv preprint arXiv:1802.06984*, 2018.

[14] Vassilios V Digalakis, Dimitry Rtischev, and Leonardo G Neumeyer, "Speaker adaptation using constrained estimation of Gaussian mixtures," *IEEE Trans. Speech Audio Process*, vol. 3, no. 5, pp. 357–366, 1995.

[15] Mark J.F Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech & Language*, vol. 12, no. 2, pp. 75–98, 1998.

[16] Junichi Yamagishi, Takao Kobayashi, Yuji Nakano, Katsumi Ogata, and Juri Isogai, "Analysis of speaker adaptation algorithms for HMM-based speech synthesis and a constrained SMAPLR adaptation algorithm," *IEEE Trans. Audio, Speech, Language Process*, vol. 17, no. 1, pp. 66–83, 2009.

[17] Yuchen Fan, Yao Qian, Frank K Soong, and Lei He, "Multi-speaker modeling and speaker adaptation for DNN-based TTS synthesis," in *Proc. ICASSP*, 2015, pp. 4475–4479.

[18] Zhiying Huang, Heng Lu, Ming Lei, and Zhijie Yan, "Linear networks based speaker adaptation for speech synthesis," in *Proc. ICASSP*, 2018, pp. 5319–5323.

[19] Nobukatsu Hojo, Yusuke Ijima, and Hideyuki Mizuno, "DNN-based speech synthesis using speaker codes," *IEICE T. Inf. Syst.*, vol. 101, no. 2, pp. 462–472, 2018.

[20] Lahiru Samarakoon and Khe Chai Sim, "Factorized hidden layer adaptation for deep neural network based acoustic modeling," *IEEE/ACM Trans. Audio, Speech, Language Process*, vol. 24, no. 12, pp. 2241–2250, 2016.

[21] Yong Zhao, Jinyu Li, Kshitiz Kumar, and Yifan Gong, "Extended low-rank plus diagonal adaptation for deep and recurrent neural networks," in *Proc. ICASSP*, 2017, pp. 5040–5044.

[22] Pawel Swietojanski and Steve Renals, "Learning hidden unit contributions for unsupervised speaker adaptation

of neural network acoustic models," in *Proc. SLT*, 2014, pp. 171–176.

[23] Sercan O Arik, Jitong Chen, Kainan Peng, Wei Ping, and Yanqi Zhou, "Neural voice cloning with a few samples," *arXiv preprint arXiv:1802.06006*, 2018.

[24] Rama Doddipatla, Norbert Braunschweiler, and Ranniery Maia, "Speaker adaptation in DNN-based speech synthesis using d-vectors," in *Proc. Interspeech*, 2017, pp. 3404–3408.

[25] RJ Skerry-Ryan, Eric Battenberg, Ying Xiao, Yuxuan Wang, Daisy Stanton, Joel Shor, Ron J Weiss, Rob Clark, and Rif A Saurous, "Towards end-to-end prosody transfer for expressive speech synthesis with Tacotron," *arXiv preprint arXiv:1803.09047*, 2018.

[26] Nagendra Kumar and Andreas G Andreou, "Heteroscedastic discriminant analysis and reduced rank HMMs for improved speech recognition," *Speech Commun.*, vol. 26, no. 4, pp. 283–297, 1998.

[27] Daniel Povey, Lukáš Burget, Mohit Agarwal, Pinar Akyazi, Feng Kai, Arnab Ghoshal, Ondřej Glembek, Nagendra Goel, Martin Karafiát, Ariya Rastrow, et al., "The subspace Gaussian mixture modela structured model for speech recognition," *Computer Speech & Language*, vol. 25, no. 2, pp. 404–439, 2011.

[28] Xiaodong Cui, Vaibhava Goel, and George Saon, "Embedding-based speaker adaptive training of deep neural networks," *arXiv preprint arXiv:1710.06937*, 2017.

[29] Lahiru Samarakoon, Brian Mak, and Khe Chai Sim, "Learning factorized transforms for unsupervised adaptation of LSTM-RNN acoustic models," in *Proc. Interpeech*, 2017, pp. 774–748.

[30] Jian Xue, Jinyu Li, and Yifan Gong, "Restructuring of deep neural network acoustic models with singular value decomposition.," in *Interspeech*, 2013, pp. 2365–2369.

[31] Jian Xue, Jinyu Li, Dong Yu, Mike Seltzer, and Yifan Gong, "Singular value decomposition based low-footprint speaker adaptation and personalization for deep neural network," in *Proc. ICASSP*, 2014, pp. 6359–6363.

[32] Christophe Veaux, Junichi Yamagishi, and Simon King, "The voice bank corpus: Design, collection and data analysis of a large regional accent speech database," in *Proc. O-COCOSDA/CASLRE*, 2013, pp. 1–4.

[33] Christophe Veaux, Junichi Yamagishi, and Kirsten MacDonald, "CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit," 2017, http://dx.doi.org/10.7488/ds/1994.

[34] Hieu-Thi Luong and Junichi Yamagishi, "Multimodal speech synthesis architecture for unsupervised speaker adaptation," in *Proc. Interspeech*, 2018, pp. 2494–2498.

[35] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[36] Tomoki Hayashi, Akira Tamamori, Kazuhiro Kobayashi, Kazuya Takeda, and Tomoki Toda, "An investigation of multi-speaker training for wavenet vocoder," in *Proc. ASRU*, 2017, pp. 712–718.

[37] Masanori Morise, Fumiya Yokomori, and Kenji Ozawa, "World: a vocoder-based high-quality speech synthesis system for real-time applications," *IEICE T. Inf. Syst.*, vol. 99, no. 7, pp. 1877–1884, 2016.

[38] Yi-Chiao Wu, Kazuhiro Kobayashi, Tomoki Hayashi, Patrick Lumban Tobing, and Tomoki Toda, "Collapsed speech segment detection and suppression for wavenet vocoder," in *Proc. Interspeech*, 2018, pp. 1988–1992.

[39] Xin Wang, Shinji Takaki, and Junichi Yamagishi, "Investigating very deep highway networks for parametric speech synthesis," *Speech Commun.*, vol. 96, pp. 1–9, 2018.

[40] Li-Juan Liu, Zhen-Hua Ling, Yuan Jiang, Ming Zhou, and Li-Rong Dai, "Wavenet vocoder with limited training data for voice conversion," in *Proc. Interspeech*, 2018, pp. 1983–1987.

[41] Berrak Sisman, Mingyang Zhang, and Haizhou Li, "A voice conversion framework with tandem feature sparse representation and speaker-adapted wavenet vocoder," in *Proc. Interspeech*, 2018, pp. 1978–1982.