Perceptual-Similarity-Aware Deep Speaker Representation Learning for Multi-Speaker Generative Modeling

Yuki Saito, *Student Member, IEEE*, Shinnosuke Takamichi, *Member, IEEE*, and Hiroshi Saruwatari, *Member, IEEE*

***Abstract*—We propose novel deep speaker representation learning that considers perceptual similarity among speakers for multi-speaker generative modeling. Following its success in accurate discriminative modeling of speaker individuality, knowledge of deep speaker representation learning (i.e., speaker representation learning using deep neural networks) has been introduced to multi-speaker generative modeling. However, the conventional discriminative algorithm does not necessarily learn speaker embeddings suitable for such generative modeling, which may result in lower quality and less controllability of synthetic speech. We propose three representation learning algorithms that utilize a perceptual speaker similarity matrix obtained by large-scale perceptual scoring of speaker-pair similarity. The algorithms train a speaker encoder to learn speaker embeddings with three different representations of the matrix: a set of vectors, the Gram matrix, and a graph. Furthermore, we propose an active learning algorithm that iterates the perceptual scoring and speaker encoder training. To obtain accurate embeddings while reducing costs of scoring and training, the algorithm selects unscored speaker-pairs to be scored next on the basis of the sequentially-trained speaker encoder's similarity prediction results. Experimental evaluation results show that 1) the proposed representation learning algorithms learn speaker embeddings strongly correlated with perceptual speaker-pair similarity, 2) the embeddings improve synthetic speech quality in speech autoencoding tasks better than conventional d-vectors learned by discriminative modeling, 3) the proposed active learning algorithm achieves higher synthetic speech quality while reducing costs of scoring and training, and 4) among the proposed similarity {vector, matrix, graph} embedding algorithms, the first achieves the best speaker similarity for synthetic speech and the third gives the most improvement in the synthetic speech naturalness.**

***Index Terms*—Deep speaker representation learning, active learning, multi-speaker generative modeling, perceptual speaker similarity, speaker embedding.**

I. Introduction

DEEP speaker representation learning is a technology for training a deep neural network (DNN)-based speaker encoder that extracts speaker embeddings (i.e., distributed representations of speakers) from input speech [1]. Traditionally, speaker embeddings have contributed to improve accuracy in discriminative modeling of speaker individuality, such as speaker recognition [2] and verification [3]. A typical training algorithm for a speaker encoder is based on speaker classification using speaker embeddings, which enables the embeddings to discriminate speaker identity of input speech accurately. A d-vector [4] and an x-vector [5] are well-known examples of speaker embeddings learned by such a speaker-discriminative training algorithm. These DNN-based speaker embeddings achieve higher accuracy in speaker recognition and verification tasks than conventional i-vectors [6].

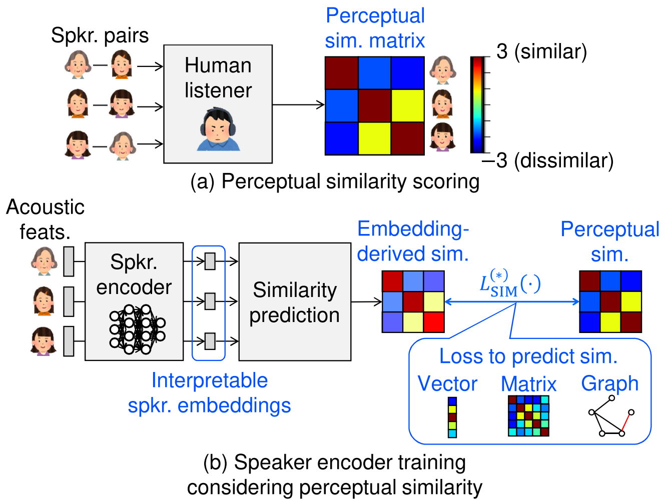
Following the success in the *discriminative* modeling of speaker individuality, knowledge of deep speaker representation learning has been transferred to *generative* modeling of speech, such as statistical text-to-speech (TTS) synthesis [7], [8] and voice conversion (VC) [9], [10]. Single-speaker high-quality generative modeling has been accomplished thanks to the developments of DNN-based speech waveform modeling [11], [12], rich acoustic models (e.g., Tacotron [13], FastSpeech [14], [15], and MelNet [16]), and sophisticated acoustic model training (e.g., generative adversarial network (GAN)-based methods [17], [18]). Speaker embeddings can advance the *single-speaker* generative modeling to a *multi-speaker* one that can synthesize any of the seen speakers' voices using a single generative model. A d-vector, for instance, acts as auxiliary input to the generative model for controlling the speaker individuality of synthetic speech [19]–[21]. In addition, multi-speaker generative modeling has the capability to synthesize unseen speakers' voices, which is very attractive because it increases the diversity of speaker individuality and widens the range of speech synthesis applications (e.g., speaker anonymization [22] and data augmentation for speech recognition [23]).

Another important factor in multi-speaker generative modeling is *intuitiveness* in controlling speaker individuality. A speaker embedding in multi-speaker generative modeling is desirable to represent a *perceptual relationship among multiple speakers* as well as an *identity of a single speaker*, i.e., the more similar speakers' voices sound, the closer their speaker embeddings positions in a speaker embedding space. Such human-perception-oriented embedding space enables a user to explore the space to find his/her favorite voice characteristics easily

Manuscript received September 4, 2020; revised December 28, 2020 and February 9, 2021; accepted February 9, 2021. Date of publication February 15, 2021; date of current version March 10, 2021. This work was supported in part by the JSPS KAKENHI under Grants 19H01116, 18K18100, 18J22090 and 17H06101, and in part by the MIC/SCOPE #182103104. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Heiga Zen. *(Corresponding author: Yuki Saito.)*

The authors are with the Graduate School of Information Science and Technology, The University of Tokyo, Bunkyo-ku, Tokyo 113-8656, Japan (e-mail: yuuki_saito@ipc.i.u-tokyo.ac.jp; shinnosuke_takamichi@ipc.i.u-tokyo.ac.jp; hiroshi_saruwatari@ipc.i.u-tokyo.ac.jp).

Digital Object Identifier 10.1109/TASLP.2021.3059114



(a) Perceptual similarity scoring

(b) Speaker encoder training considering perceptual similarity

Fig. 1. Conceptual diagram of proposed deep speaker representation learning. We first conduct (a) perceptual similarity scoring to obtain a matrix representing the perceptual relationships among speakers. We then perform (b) speaker encoder training to predict the relationships from the speaker embeddings. As a result, similar speakers position are positioned closer to each other and dissimilar ones are kept away in the speaker embedding space.

(c.f., interactive visual design optimization based on a preferential Bayesian optimization [24], [25]). Also, the introduction of perceptual similarity among speakers has the possibility in reproducing an unseen speaker's voice characteristics from a few of his/her speech utterances (i.e., speaker adaptation of a speech synthesis model [26], [27]) because information of seen speakers similar to him/her can enable the speaker adaptation to function effectively. A conventional speaker-discriminative learning of speaker embeddings completely ignores perceptual similarity among speakers, resulting in a decrease in the controllability and adaptability of multi-speaker generative modeling [28]. For example, a multi-speaker TTS synthesis with a Tacotron-based acoustic model and a pretrained d-vector extractor [21] can synthesize significantly high-quality speech, but the speaker similarity of the synthetic speech is somewhat degraded when the speakers are unseen during the DNN training.

To learn better speaker embeddings suitable for multi-speaker generative modeling, we propose deep speaker representation learning algorithms that consider perceptual speaker similarity among speakers. Fig. 1 illustrates a conceptual diagram of the proposed deep speaker representation learning. We first conduct large-scale scoring of perceptual speaker-pair similarity to obtain a *perceptual speaker similarity matrix*. We then train a speaker encoder to minimize a loss function defined by the similarity matrix. We proposed 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 embedding. Similarity *matrix* embedding directly utilizes the whole similarity matrix as the target to be predicted by a speaker encoder and trains it to minimize the Frobenius norm between the Gram matrix of a set of speaker embeddings (i.e., embedding-derived speaker similarity matrix) 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 embeddings. Furthermore, we propose an active learning algorithm that iterates the perceptual speaker-pair similarity scoring and speaker encoder training, which aims to reduce the number of scoring times that quadratically increases with that of seen speakers. To obtain accurate embeddings while reducing scoring and training costs, the algorithm selects unscored speaker-pairs to be scored next on the basis of the sequentially-trained speaker encoder's similarity prediction results. Fig. 2 shows the relationship between the conventional and proposed approaches to deep speaker representation learning.

In experimental evaluations, we first conduct large-scale scoring, and then evaluate the proposed speaker representation algorithms and active learning algorithm. The evaluation results show that 1) the proposed speaker representation learning algorithms learn speaker embeddings strongly correlated with perceptual similarity scores, 2) the embeddings improve synthetic speech quality in speech autoencoding tasks compared with conventional d-vectors obtained by a speaker-discriminative learning algorithm, 3) the proposed active learning algorithm achieves higher synthetic speech quality while reducing costs of scoring and training, and 4) among the proposed similarity {vector, matrix, graph} embedding algorithms, the first achieves the best speaker similarity for synthetic speech, and the third gives the highest AUC value for similar speaker-pair detection and the most improvement in the synthetic speech naturalness.

This paper is organized as follows. Section II briefly reviews a conventional discriminative approach to deep speaker representation learning. Section III describes the three components in our approach: scoring perceptual speaker-pair similarity, deep speaker representation learning considering the similarity, and active learning. Section IV presents experimental evaluations. Section V concludes this paper.

Note that this paper is partially based on an international conference paper written by the authors [29]. The additional contributions of this paper are the introduction of graph embedding technology [30] and an active learning framework [31] to the proposed perceptual-similarity-aware deep speaker representation learning.

II. Conventional Speaker-Classification-Based Deep Speaker Representation Learning

This section holds a d-vector up as an example and describes conventional deep speaker representation based on a speaker classification task. A learned d-vector can be used for conditioning a multi-speaker generative model to control the speaker individuality of synthetic speech.

A. D-Vector

A d-vector [4] is a bottleneck feature vector extracted from a speaker encoder that performs feature extraction in a DNN-based speaker classification model. The DNNs take an acoustic feature sequence as input and predict a one-hot speaker code $\boldsymbol{c} = [c(1), \ldots, c(n), \ldots, c(N_{\mathrm{s}})]^\top$ that represents the identity of

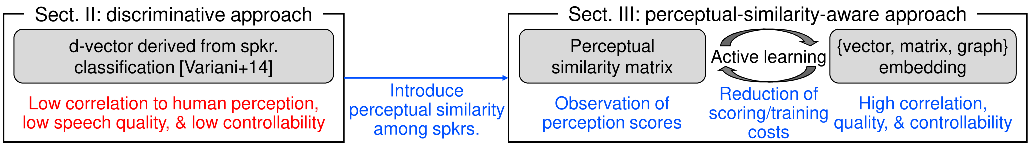

Sect. II: discriminative approach
d-vector derived from spkr. classification [Variani+14]
Low correlation to human perception, low speech quality, & low controllability
Introduce perceptual similarity among spkrs.
Sect. III: perceptual-similarity-aware approach
Perceptual similarity matrix
Active learning
{vector, matrix, graph} embedding
Observation of perception scores
Reduction of scoring/training costs
High correlation, quality, & controllability

Fig. 2. Relationship between conventional deep speaker representation learning approach and proposed one.

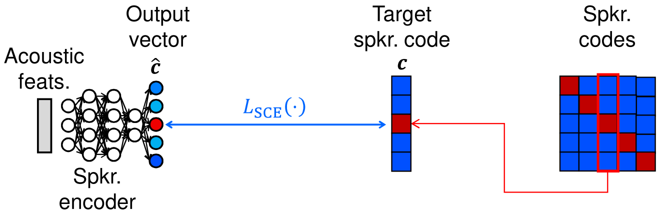


Fig. 3. Conventional speaker encoder training based on speaker classification. d-vector $\boldsymbol{d}$ is an output of a squeeze layer immediately before the output layer.

one of the seen (i.e., pre-stored) N_{s} speakers. The ith speaker's identity $\boldsymbol{c}_i$ is defined as follows:

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

A loss function for the DNN training is defined as the softmax cross-entropy of speaker classification:

$$L_{\mathrm{SCE}}\left(\boldsymbol{c}, \hat{\boldsymbol{c}}\right) = -\sum_{n=1}^{N_{\mathrm{s}}} c(n) \log \hat{c}(n), \tag{2}$$

where $\hat{\boldsymbol{c}} = [\hat{c}(1), \ldots, \hat{c}(n), \ldots, \hat{c}(N_{\mathrm{s}})]^\top$ is an output vector of the DNNs. Fig. 3 shows the computation procedure of $L_{\mathrm{SCE}}(\cdot)$. This loss function is calculated frame by frame, and minibatch stochastic gradient descent (SGD) is applied to update the model parameters of the speaker encoder.

After the training, an N_{d}-dimensional d-vector $\boldsymbol{d} = [d(1), \ldots, d(N_{\mathrm{d}})]^\top$ 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} to use the lower-dimensional speaker embedding. The ith speaker's d-vector $\boldsymbol{d}_i$ is extracted from the speaker's acoustic feature sequences and averaged across all frames in all utterances.

B. Multi-Speaker Generative Modeling Using D-Vectors

In multi-speaker generative modeling, a single generative model is trained to synthesize multiple speakers' voices [32]. A speaker embedding is fed into the generative model to control speaker identity of synthetic speech [19], [20]. A well-trained multi-speaker generative model can even synthesize an unseen speaker's voice using a small amount of his/her speech data to adapt the model parameters (i.e., weights and bias of DNNs) or the speaker embedding that can well reproduce the speaker's voice characteristics.

One can use a per-speaker d-vector for multi-speaker generative modeling [19], [20] as shown in Fig. 4. Thanks to its low-dimensional continuous representation, the use of the d-vector shows better performance in speaker adaptation than that of a simple one-hot speaker code [33]. However, the speaker-classification-based speaker embeddings have not only less interpretability but also the possibility of worsening synthetic speech quality in speaker adaptation [28].

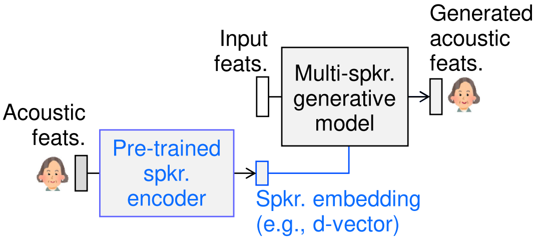

Input feats.
Multi-spkr. generative model
Generated acoustic feats.
Acoustic feats.
Pre-trained spkr. encoder
Spkr. embedding (e.g., d-vector)

Fig. 4. Multi-speaker generative modeling using speaker embeddings. Pre-trained speaker encoder extracts a per-speaker embedding from acoustic features in advance, and a multi-speaker generative model predicts acoustic features from input features. Speaker embeddings control speaker individuality of synthetic speech.

III. Proposed Perceptual-Similarity-Aware Deep Speaker Representation Learning

As we mentioned in Section I, the introduction of a human's speaker-similarity perception into deep speaker representation learning has the possibility of improving controllability and adaptability of a multi-speaker generative model. In this section, we first introduce a speaker similarity matrix obtained by perceptual scoring, which becomes the target of speaker encoder training. Then, we propose speaker representation learning algorithms that predict the perceptual scores from acoustic features. To reduce scoring and training costs, we finally propose an active learning algorithm.

A. Perceptual Speaker Similarity Matrix

We define a perceptual speaker similarity matrix that represents the pairwise speaker similarity perceived by listeners. Let $\mathbf{S} = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_i, \ldots, \boldsymbol{s}_{N_{\mathrm{s}}}]$ be an N_{s}-by-N_{s} symmetric similarity matrix and $\boldsymbol{s}_i = [s_{i,1}, \ldots, s_{i,j}, \ldots, s_{i,N_{\mathrm{s}}}]^\top$ 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. We define $s_{i,j}$ as the average score of perceptual scoring that asks listeners "To what degree do the ith speaker's voice and the jth speaker's one sound similar? Please answer the degree of similarity as a value between $-v$ and v." To focus on modeling the inter-speaker perceptual similarity, we exclude same-speaker pairs from the scoring; further, we assume that the diagonal elements $s_{i,i}$, i.e., intra-speaker perceptual similarity, take the maximum value v. Fig. 5 illustrates the perceptual scoring process. Fig. 6(a) and (b) show a perceptual speaker similarity matrix of 153 female

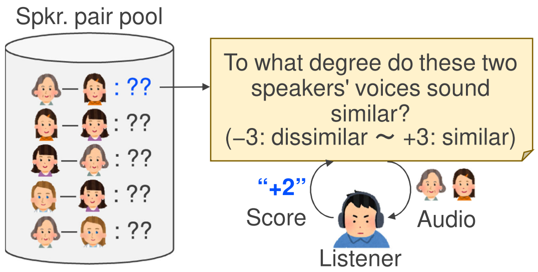

Spkr. pair pool
: ??
: ??
: ??
: ??
: ??
To what degree do these two speakers' voices sound similar?
(–3: dissimilar ∼ +3: similar)
"+2"
Score
Audio
Listener

Fig. 5. Perceptual scoring of speaker-pair similarity. Speaker-pair pool stores the speaker pairs to be scored. Listener is asked to score perceptual similarity of two presented speakers' voices as an integer between $-v$ and v. In this figure, $v = 3$.

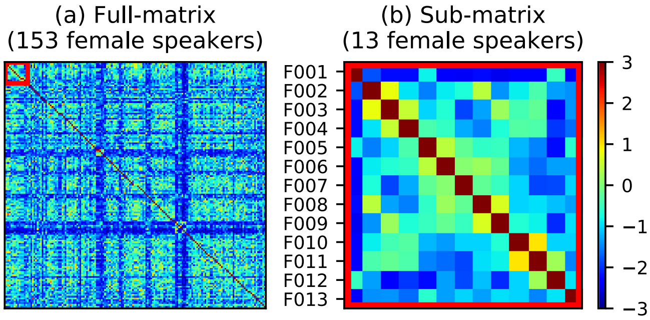


Fig. 6. (a) Perceptual speaker similarity matrix of 153 female Japanese speakers obtained by large-scale perceptual scoring and (b) its sub-matrix.

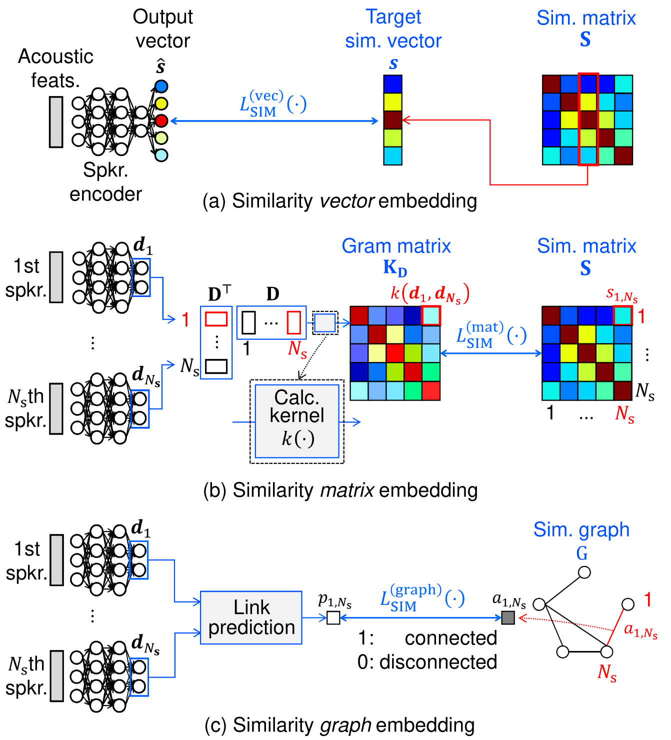


Fig. 7. Calculation of loss functions in proposed algorithms based on (a) similarity vector embedding, (b) similarity matrix embedding, and (c) similarity graph embedding.

Japanese speakers and its sub-matrix, respectively.[1] Please see Section IV-A1 for details of the perceptual scoring and Section IV-B for the analysis results of the scores.

B. Perceptual-Similarity-Aware Deep Speaker Representation Learning Algorithms

We proposed three algorithms for learning similarity-aware speaker embeddings with different representations of the perceptual speaker similarity matrix: a set of similarity vectors, the Gram matrix, and a graph.

1) 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:

$$L^{(\mathrm{vec})}_{\mathrm{SIM}}(\boldsymbol{s}, \hat{\boldsymbol{s}}) = \frac{1}{N_\mathrm{s}} (\hat{\boldsymbol{s}} - \boldsymbol{s})^\top (\hat{\boldsymbol{s}} - \boldsymbol{s}), \tag{3}$$

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. Fig. 7(a) shows the computation procedure of $L^{(\mathrm{vec})}_{\mathrm{SIM}}(\cdot)$. This loss function is calculated frame by frame, and minibatch SGD is applied to update the model parameters of the speaker encoder.

2) Similarity Matrix Embedding: The second algorithm directly uses a perceptual speaker similarity matrix as a constraint on coordinates of speaker embeddings. Let $\mathbf{D} = [\boldsymbol{d}_1, \ldots, \boldsymbol{d}_i, \ldots, \boldsymbol{d}_{N_\mathrm{s}}]$ be an N_d-by-N_s matrix including speaker embeddings extracted from all seen speakers. A loss function for the training is defined as follows:

$$L^{(\mathrm{mat})}_{\mathrm{SIM}}(\mathbf{D}, \mathbf{S}) = \frac{2}{\|\mathbf{1}_{N_\mathrm{s}} - \mathbf{I}_{N_\mathrm{s}}\|_F^2} \left\| \widetilde{\mathbf{K}}_\mathbf{D} - \widetilde{\mathbf{S}} \right\|_F^2, \tag{4}$$

$$\widetilde{\mathbf{K}}_\mathbf{D} = \mathbf{K}_\mathbf{D} - (\mathbf{K}_\mathbf{D} \odot \mathbf{I}_{N_\mathrm{s}}), \tag{5}$$

$$\widetilde{\mathbf{S}} = \mathbf{S} - v\mathbf{I}_{N_\mathrm{s}}, \tag{6}$$

where $\|\cdot\|_F$, $\odot$, $\mathbf{1}_{N_\mathrm{s}}$, and $\mathbf{I}_{N_\mathrm{s}}$ denote the Frobenius norm of a given matrix, the Hadamard product, an N_s-by-N_s matrix whose components are all 1, and the N_s-by-N_s identity matrix, respectively. $2/\|\mathbf{1}_{N_\mathrm{s}} - \mathbf{I}_{N_\mathrm{s}}\|_F^2$ is a normalization coefficient corresponding to the degrees of freedom of the matrix $\widetilde{\mathbf{K}}_\mathbf{D} - \widetilde{\mathbf{S}}$. $\mathbf{K}_\mathbf{D}$ is the Gram matrix of a set of speaker embeddings defined as:

$$\mathbf{K}_\mathbf{D} = \begin{bmatrix} k(\boldsymbol{d}_1, \boldsymbol{d}_1) & \cdots & k(\boldsymbol{d}_1, \boldsymbol{d}_{N_\mathrm{s}}) \\ \vdots & \ddots & \vdots \\ k(\boldsymbol{d}_{N_\mathrm{s}}, \boldsymbol{d}_1) & \cdots & k\left(\vec{d}_{N_\mathrm{s}}, \vec{d}_{N_\mathrm{s}}\right) \end{bmatrix}, \tag{7}$$

where $k(\boldsymbol{d}_i, \boldsymbol{d}_j)$ is a kernel function of a pair of speaker embeddings $\boldsymbol{d}_i$ and $\boldsymbol{d}_j$, i.e., speaker-embedding-derived speaker similarity. The choice of $k(\boldsymbol{d}_i, \boldsymbol{d}_j)$ depends on how we normalize the similarity scores $s_{i,j}$ during the training. For example, if we normalize the scores to be in $[-1, +1]$, the sigmoid kernel $k(\boldsymbol{d}_i, \boldsymbol{d}_j) = \tanh(\boldsymbol{d}_i^\top \boldsymbol{d}_j)$ is a possible choice. This proposed algorithm therefore makes the speaker-embedding-derived

[1] We also conducted perceptual similarity scoring among the 13 speakers ("F001"–"F013"), including the same-speaker pairs. However, we have not used those results for any experimental evaluations in this study, because we aimed to focus on perceptual similarity modeling among different speakers. Appendix A summarizes and discusses those scoring results.

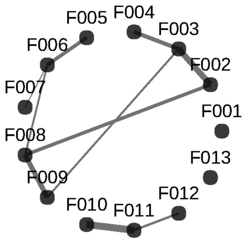


Fig. 8. Speaker similarity graph defined by similarity matrix shown in Fig. 6(b). Each node represents one speaker, and links connect perceptually similar pairs (i.e., $s_{i,j} > 0$). The wider the links are, the more similar speaker pairs are.

speaker similarity closer to the perceptual speaker similarity. Fig. 7(b) shows the computation procedure of $L_{\mathrm{SIM}}^{(\mathrm{mat})}(\cdot)$. This loss function is calculated during the training as follows. First, fixed-length (e.g., 256 frames) acoustic feature sequences for all seen speakers are randomly sampled from the training dataset to construct a minibatch. Second, per-speaker embeddings are extracted from each acoustic feature sequence in the minibatch and averaged across the sequence. Third, the Gram matrix (Eq. (7)) is calculated using the embeddings. Finally, the loss function (Eq. (4)) is calculated, and minibatch SGD is applied to update the speaker encoder's model parameters.

3) Similarity Graph Embedding: The third algorithm is a variant of the second algorithm and learns relationships 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. 8. We define an N_{s}-by-N_{s} adjacency matrix $\mathbf{A}$ that determines the existence of the links on the basis of elements of the similarity matrix $\mathbf{S}$. In this paper, we use a soft adjacency matrix; i.e., each element $a_{i,j}$ takes a real value between 0 (disconnected) and 1 (connected), which is calculated as $a_{i,j} = (s_{i,j} + v)/2v$. A loss function for the training is defined as follows:

$$L_{\mathrm{SIM}}^{(\mathrm{graph})}(\mathbf{D}, \mathbf{A}) = -\sum_{i,j=1, i\neq j}^{N_{\mathrm{s}}} a_{i,j} \log p_{i,j} - \sum_{i,j=1, i\neq j}^{N_{\mathrm{s}}} (1 - a_{i,j}) \log (1 - p_{i,j}), \quad (8)$$

where $p_{i,j}$ denotes a link probability, which is defined as $p_{i,j} = \exp(-||\boldsymbol{d}_i - \boldsymbol{d}_j||_2^2)$ referring to [34]. Fig. 7(c) shows the computation procedure of $L_{\mathrm{SIM}}^{(\mathrm{graph})}(\cdot)$. The calculation procedure in this loss function—i.e., the extraction and aggregation of the speaker embedding—is similar to the one described in Section III-B2.

C. Multi-Speaker Generative Modeling Using Perceptual-Similarity-Aware Speaker Embeddings

We can easily apply the proposed perceptual-similarity-aware speaker embeddings to multi-speaker generative modeling by replacing a speaker encoder in Fig. 4 with one trained by the proposed algorithms. The embeddings can be expected to improve synthetic speech quality and controllability of the modeling since they are learned to predict similarity scores that represent perceptual relationships among speakers, not speaker codes that completely ignore the relationships.

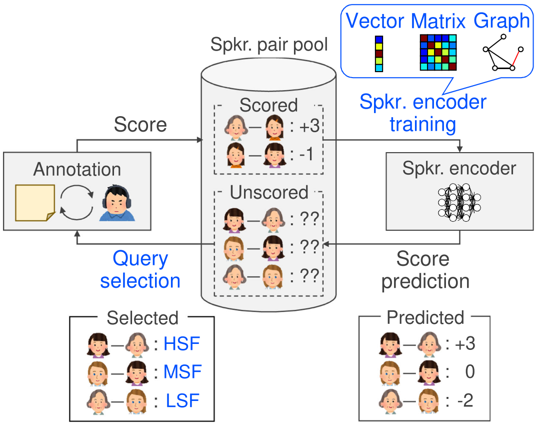


Fig. 9. Active learning of perceptual-similarity-aware speaker embeddings.

D. Active Learning of Perceptual-Similarity-Aware Speaker Embeddings

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. We propose an active learning algorithm to reduce the scoring and training costs. Active learning [31] is a general framework to sequentially train a machine learning model with a small labeled dataset and large unlabeled one, which iterates 1) model training with the labeled dataset and 2) query selection to increase labeled data.

Fig. 9 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.

1) 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. (3), (4), or (8). Note that the speaker encoder's model parameters are not reset at every active learning iteration.

2) 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. We investigate three query strategies: 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) middle-similarity first (MSF) that selects a speaker pair whose predicted similarity is closer to 0.

E. Discussion

Regarding prior work, Tachibana *et al.* [35] and Ohta *et al.* [36] proposed controllable speech synthesis in the hidden Markov model (HMM) and Gaussian mixture model (GMM) era. They modeled a speaker's voice characteristics with a pair of subjective impression words such as "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 relationship between a speaker's intention and listener's perception (e.g., difference in emotion perception [37]) by using the algorithms. Also, we can use the proposed speaker embeddings in more sophisticated speech synthesis frameworks, such as end-to-end multi-speaker TTS [21], multi-speaker multi-lingual TTS [38], and singing VC [39], instead of the conventional discriminative speaker embeddings.

The similarity vector embedding in Section III-B1 can train a speaker encoder with a criterion that is simpler than the other two proposed algorithms, which predict the whole similarity matrix or the similarity graph structure. However, it is not very flexible in handling an increased number of seen speakers, because the dimensionality of the output layer is fixed.

Both the similarity matrix and graph embedding algorithms can directly learn the relationships among speakers via the matrix or graph. The difference between them is the approach to optimization: the former is regression based, while the latter is classification based. In Section IV, we empirically show that the classification-based approach can improve the naturalness of synthetic speech and work well in the proposed active learning setting compared with the regression-based one.

In the Gram matrix calculation of the similarity matrix embedding, we can choose an arbitrary kernel function to construct a speaker embedding space. When we use the inner product as the kernel function, Eq. (4) is equivalent to deep clustering [40] (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 III-B3 introduces knowledge of graph embedding [30] to deep speaker representation learning. One can further incorporate graph signal processing [41] and graph neural networks [42] to the algorithm for better modeling.

Active learning in Section III-D can be regarded as human-in-the-loop (HITL) learning [43] of speaker embeddings considering human speech perception. From this viewpoint, one can extend the HITL learning framework to speech synthesis that takes a human listener's speech quality assessment into account for the model training (e.g., the GAN training incorporating a human-based discriminator [44]).

IV. Experimental Evaluation

A. Experimental Conditions

1) Conditions for Large-Scale Perceptual Scoring: We conducted large-scale perceptual scoring to obtain the similarity matrix $\mathbf{S}$. We used 153 female Japanese speakers from the JNAS corpus [45]. Each speaker utters at least 150 reading-style utterances (totaling about 44 hours). We extracted five non-parallel utterances per speaker to score 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). We recruited listeners using Lancers,[2] a well-known crowdsourcing platform in Japan. At least 10 different listeners scored the similarity of each of the 11 628 speaker pairs. The total numbers of listeners and answers were 4060 and 138 040, respectively.

2) Conditions for Deep Speaker Representation Learning: We used the JNAS corpus to train a DNN-based speaker encoder and assumed that the 13 speakers shown in Fig. 6(b) (from "F001" to "F013") were unseen during the training. In the training, we used 90% of the remaining 140 seen speakers' utterances and balanced the number of utterances per speaker. In the evaluation, we used the unseen speakers' 50 utterances and the seen speakers' remaining ones. We omitted the five utterances for the perceptual scoring from both the training and evaluation data.

During the training, we normalized each element in the similarity matrix to be in $[-1, +1]$ for the similarity vector or matrix embedding (Sections III-B1 and III-B2) and in [0, 1] for the similarity graph embedding (Section III-B3) by a linear transformation: $s_{i,j}/3$ in the former case and $(s_{i,j}+3)/6$ in the latter case. Accordingly, we used the sigmoid kernel $k(\boldsymbol{d}_i, \boldsymbol{d}_j) = \tanh(\boldsymbol{d}_i^\top \boldsymbol{d}_j)$ for the Gram matrix calculation in Eq. (7). Note that the sigmoid kernel is not positive definite, and other choices such as the cosine similarity and Gaussian kernels are also available for the proposed algorithm. In the similarity graph embedding, we defined the adjacency matrix using the normalized similarity matrix that represents the likelihood of a link existence as a value between [0, 1]. We used 256 frames of acoustic feature sequence to calculate per-speaker embeddings in the proposed similarity matrix or graph embedding algorithm.

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 embedding extraction were 256 and 8, respectively. In the d-vector learning (Section II-A) and the similarity vector embedding (Section III-B1), we prepared an output layer with 140 units whose 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. We used the STRAIGHT vocoder [47] to extract the mel-cepstral coefficients and normalized them to have a zero-mean and unit-variance during the training. The optimization algorithm was AdaGrad [48], setting its learning rate to 0.01. The number of epochs for the training was 100.

3) Conditions for Multi-Speaker Generative Modeling: We constructed a variational autoencoder (VAE)-based multi-speaker generative model [20] that incorporates a DNN-based speech recognition model and a speaker encoder into speech

[2] https://www.lancers.jp

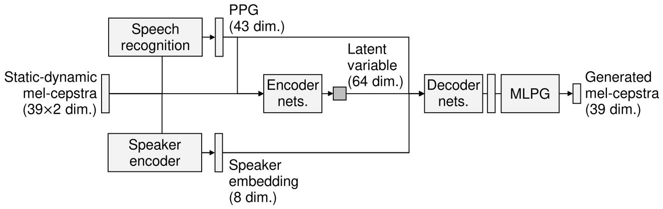

Speech recognition
PPG (43 dim.)
Latent variable (64 dim.)
Static-dynamic mel-cepstra (39×2 dim.)
Encoder nets.
Decoder nets.
MLPG
Generated mel-cepstra (39 dim.)
Speaker encoder
Speaker embedding (8 dim.)

Fig. 10. VAE-based multi-speaker generative model used in the experimental evaluation. Here, "MLPG" means maximum likelihood parameter generation [46] using the static-dynamic features predicted from the decoder networks. The speaker embedding is predicted from the input static-dynamic mel-cepstral coefficient sequence by using the speaker encoder and averaged across all frames in the sequence.

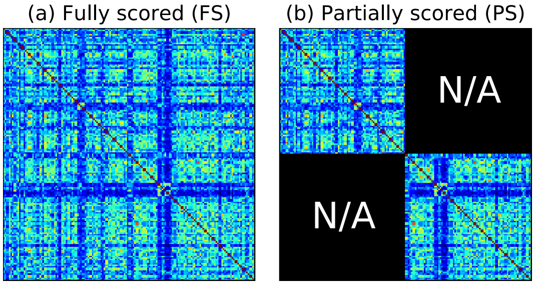


Fig. 11. (a) Fully scored (FS) and (b) partially scored (PS) settings. In the active learning evaluation (Sections IV-D1 and IV-D2), we used the "PS" setting as the starting point for all active learning setups.

synthesis. The VAEs can be expected to learn various speakers' voice characteristics through the autoencoding process with latent variable regularization. Note that the VAE-based speech autoencoding evaluation is not very close to conventional speech synthesis such as TTS and VC, but we believe that this evaluation is sufficient to investigate the effectiveness of the proposed perceptual-similarity-aware speaker embeddings. Fig. 10 illustrates the VAE-based multi-speaker generative model that we used. 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 1024. We trained the recognition model that predicted framewise 43-dimensional Japanese phonetic posteriorgrams (PPGs) [49] from the same input vector as the speaker encoder. We used at least 50 utterances for each of the 140 seen speakers for the training. The number of epochs for the training was 100. The speaker encoder was the same as the DNNs described in Section IV-A2. The DNN architecture for the VAEs was a Feed-Forward network that consisted of encoder and decoder networks. The encoder network represented a diagonal Gaussian distribution, whose mean and variance were estimated by DNNs. The encoder had two hidden layers with the rectified linear unit (ReLU) [50] activation function and predicted the framewise mean and variance of the diagonal Gaussian distribution from a joint vector of the static-dynamic mel-cepstral coefficients and PPGs to sample 64-dimensional latent variables. The first and second hidden layers had 256 and 128 hidden units, respectively. We used the standard multivariate Gaussian $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ for the latent variable's prior distribution. The decoder network represented an isotropic Gaussian distribution, whose mean was estimated by DNNs. The decoder generated the input static-dynamic mel-cepstral coefficients frame by frame from a joint vector of the latent variables, PPGs, and 8-dimensional speaker embedding. The DNN architecture for the decoder was symmetric with respect to that for the encoder. We trained the VAEs to maximize the variational lower bound of the log likelihood [51] with 25 epochs using the same training data as that used in the speaker encoder training (Section IV-A2). The optimization algorithm for the speech recognition model and VAEs was AdaGrad, setting its learning rate to 0.01. In inference, we generated any arbitrary target speaker's static-dynamic mel-cepstral coefficients by using the trained VAEs representing a Gaussian distribution with the predicted mean vector and a fixed isotropic covariance. Specifically, we first fed a speaker's static-dynamic mel-cepstral coefficients into the speech recognition model to obtain PPGs. We then inputted a joint vector of the coefficients and PPGs to the encoder networks and predicted the framewise mean and variance of the diagonal Gaussian distribution to sample the VAE latent variables.[3] Finally, we fed a joint vector of the latent variables, the PPGs, and the target speaker's embedding into the decoder networks and generated the speaker's static-dynamic mel-cepstral coefficients. We performed the maximum likelihood parameter generation [46] to generate static mel-cepstral coefficients considering their temporal dependencies. We used the generated mel-cepstral coefficients and original speech's excitation parameters (i.e., F0 and five band-aperiodicity [52], [53]) for speech waveform synthesis using the STRAIGHT vocoder systems [47].

4) Conditions for Active Learning: In active learning, we divided the 140 seen speakers into two groups (the first 70 speakers and the remaining), and assumed that speaker similarity scores among the different groups were unobserved, as shown in Fig. 11(b). We simulated this active learning by using a binary mask to exclude unobserved scores from the loss calculation. We iterated 1) the speaker encoder training using observed similarity scores with one epoch and 2) the query selection using the trained speaker encoder. We set the number of queries per iteration to 43 empirically. The number of active learning iterations was 115. Other conditions, i.e., the numbers of training/evaluation data and their details, the DNN architectures, the speech parameter extraction, and the optimization algorithm, were the same as the ones previously described in Sections IV-A2 and IV-A3.

B. Analysis of Perceptual Similarity Scores

We analyzed the perceptual similarity scores that made the similarity matrix shown in Fig. 6(a). Fig. 12 shows a histogram of all the scores. We found that approximately 70% of the scores were smaller than zero. We also created a histogram of speaker-pairwise scores of the 13 unseen speakers in Fig. 13. We observed that the score distributions of dissimilar speaker

[3] We can also use latent variables sampled from the prior $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ instead of the ones predicted by the encoder networks. Evaluation results described in Appendix B show that the experimental evaluation using the predicted VAE latent variables is sufficient to demonstrate the proposed algorithms' effectiveness, because there were no significant differences between the quality of the speech synthesized with the predicted and sampled latent variables.

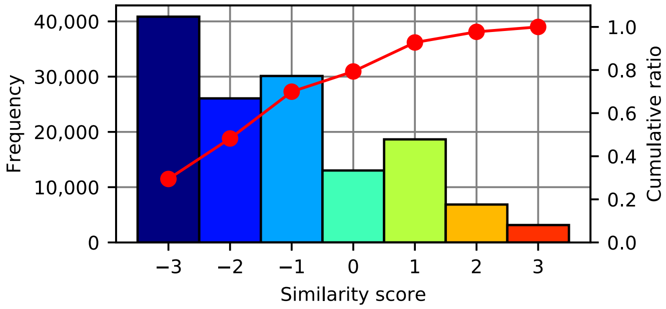


Fig. 12. Histogram of perceptual similarity scores of 153 female Japanese speakers. Red line denotes the cumulative ratio.

pairs (e.g., "F001-F009") had a lower variance than those of similar ones (e.g., "F010-F011"). These results suggested that the listeners easily found dissimilar speakers rather than similar ones. To investigate the inter-listener agreement of the whole scoring results, we calculated the Fleiss's kappa [54]. The kappa value was 0.0467, which indicated the possibility of agreement slightly better than chance. When we binarized the seven-value scale (from -3 to $+3$) into "similar" ($s_{i,j} \geq 0$) and "dissimilar" (otherwise) values, kappa increased to 0.1228. The similarity matrix we obtained is available online.[4]

C. Evaluation in Deep Speaker Representation Learning

We first evaluated whether the proposed representation learning algorithms learn speaker embeddings that consider perceptual speaker similarity and improve synthetic speech quality in multi-speaker generative modeling. We compared the following four algorithms in this section:

- **d-vec.:** Minimizing Eq. (2) [4]
- **Prop. (vec):** Minimizing Eq. (3)
- **Prop. (mat):** Minimizing Eq. (4)
- **Prop. (graph):** Minimizing Eq. (8)

1) Correlation Analysis of Speaker Embeddings: We computed 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)." Fig. 14 shows the scatter plots of the similarity scores and predicted similarity with their correlation coefficients. We found that "Prop. (*)" learned speaker embeddings that had a stronger correlation with the similarity scores than "d-vec.," which demonstrated that the proposed algorithms learned speaker embeddings considering perceptual similarity among speakers. We also observed that "Prop. (graph)" achieved the strongest correlation among the four algorithms not only in (a) "Seen-Seen" but also (b) "Seen-Unseen" speaker-pair cases, which indicated that the graph-embedding-based learning algorithm worked the best to learn pairwise relationships among speakers.

2) Performance in Similar Speaker-Pair Detection: We created a receiver operating characteristic (ROC) curve [55] of a binary classifier that detects similar speaker pairs using per-speaker embeddings. 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. Fig. 15 shows ROC curves of similar speaker-pair detection using per-speaker embeddings learned by the four different algorithms. Here, we defined "similar speaker-pair" as a pair of two speakers whose perceptual similarity is greater than 0. From this figure, we found the proposed algorithms successfully made the ROC curves closer to the upper left corner while the conventional d-vectors did not.

TABLE I
AUC VALUES OF SIMILAR SPEAKER-PAIR DETECTION USING SPEAKER EMBEDDINGS LEARNED BY FOUR DIFFERENT SPEAKER REPRESENTATION LEARNING ALGORITHMS

	d-vec.	Prop. (vec)	Prop. (mat)	Prop. (graph)
Seen-Seen	0.57	0.69	0.88	**0.92**
Seen-Unseen	0.63	0.69	0.77	**0.82**

We also calculated the area under the ROC curve (AUC) [56] that quantifies the performance of a binary classifier as a scalar value between 0.5 (random classification) and 1.0 (perfect classification). Table I shows the AUC values calculated with the ROC curves shown in Fig. 15. We found that "d-vec." resulted in the lowest AUC among the four algorithms, which suggests 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)" achieved an AUC higher than 0.8 even in the "Seen-Unseen" speaker-pair case. These results demonstrated that the proposed algorithms constructed the speaker space where we can accurately find similar speaker pairs using their embeddings.

3) Subjective Evaluation in Speaker Adaptation: We evaluated the effectiveness of the proposed speaker embeddings in speaker adaptation of the VAE-based multi-speaker generative model. In the speaker adaptation, we aimed to reconstruct the 13 unseen speakers' speech using their speaker embeddings and the trained VAEs. We conducted subjective evaluations on the naturalness and speaker similarity of the synthetic speech of the unseen speakers. We used 50 utterances of each unseen speaker for the speaker embedding extraction. We synthesized speech samples using mel-cepstral coefficients predicted by the VAEs trained with the four different speaker encoders. We evaluated the synthetic speech naturalness on the basis of a series of preference AB tests that compared the conventional algorithm ("d-vec.") with any of the three proposed algorithms ("Prop. (*)"). Twenty-five listeners participated in each of the following evaluations by using our crowdsourced evaluation system. Each listener evaluated 10 speech samples randomly extracted from the 50 utterances of each unseen speaker. Similarly, we evaluated the synthetic speech speaker similarity on the basis of a series of XAB tests 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) × 3 ("d-vec." vs. "Prop. (*)") × 13 (unseen speakers) × 25 (listeners per task set) = 1,950.

[4] http://sython.org/demo/JSPS-DC1/index.html

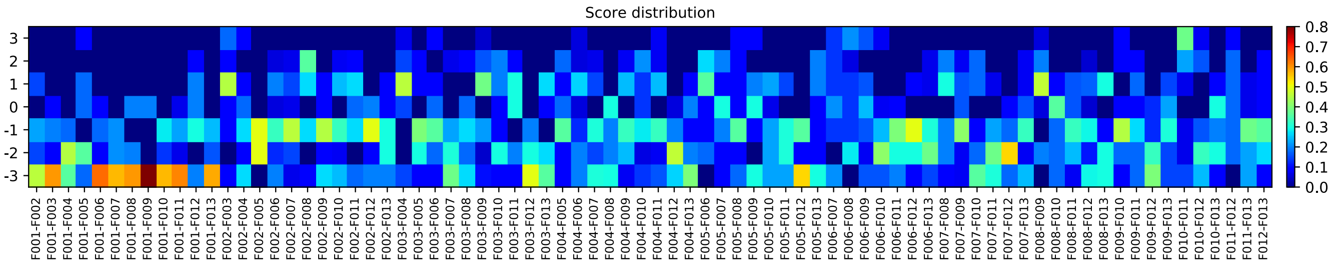


Fig. 13. Histogram of perceptual similarity scores of 13 speakers (from "F001" to "F013").

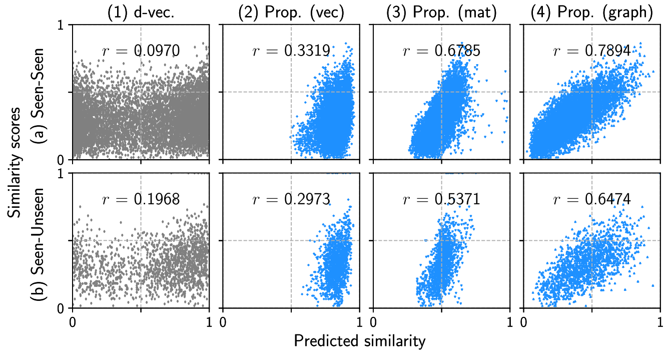


Fig. 14. Scatter plots of similarity scores and predicted similarity with their correlation coefficient r.

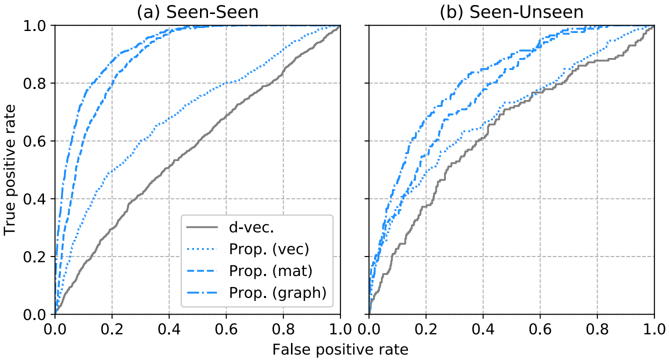


Fig. 15. ROC curves of similar speaker-pair detection using per-speaker embeddings. The closer the curve follows the upper left corner, the more accurate the speaker embeddings can detect similar speaker pairs.

TABLE II
PREFERENCE SCORES ON SYNTHETIC SPEECH NATURALNESS (LEFT: CONVENTIONAL D-VECTOR, RIGHT: PROPOSED ALGORITHM)

Speaker	Prop. (vec)	Prop. (mat)	Prop. (graph)
F001	0.41 - **0.59**	0.46 - **0.54**	0.39 - **0.61**
F002	0.46 - **0.54**	0.46 - **0.54**	0.44 - **0.56**
F003	0.42 - **0.58**	0.44 - **0.56**	0.44 - **0.56**
F004	0.45 - **0.55**	0.46 - 0.54	0.45 - **0.55**
F005	0.38 - **0.62**	0.48 - 0.52	0.43 - **0.57**
F006	0.40 - **0.60**	0.45 - **0.55**	0.45 - **0.55**
F007	0.42 - **0.58**	0.48 - 0.52	0.42 - **0.58**
F008	0.44 - **0.56**	0.38 - **0.62**	0.39 - **0.61**
F009	0.43 - **0.57**	0.49 - 0.51	0.36 - **0.64**
F010	0.44 - **0.56**	0.46 - 0.54	0.45 - **0.55**
F011	0.46 - 0.54	0.43 - **0.57**	0.31 - **0.69**
F012	0.44 - **0.56**	0.46 - 0.54	0.44 - **0.56**
F013	0.43 - **0.57**	0.44 - **0.56**	0.37 - **0.63**

TABLE III
PREFERENCE SCORES ON SYNTHETIC SPEECH SPEAKER SIMILARITY (LEFT: CONVENTIONAL D-VECTOR, RIGHT: PROPOSED ALGORITHM)

Speaker	Prop. (vec)	Prop. (mat)	Prop. (graph)
F001	0.44 - **0.56**	0.49 - 0.51	0.41 - **0.59**
F002	0.47 - 0.53	0.50 - 0.50	0.38 - **0.62**
F003	0.43 - **0.57**	0.50 - 0.50	0.44 - **0.56**
F004	0.38 - **0.62**	0.40 - **0.60**	0.45 - **0.55**
F005	0.43 - **0.57**	**0.62** - 0.38	0.43 - **0.57**
F006	0.43 - **0.57**	0.44 - **0.56**	0.38 - **0.62**
F007	0.49 - 0.51	**0.57** - 0.43	0.45 - **0.55**
F008	0.42 - **0.58**	0.50 - 0.50	0.40 - **0.60**
F009	0.40 - **0.60**	0.50 - 0.50	0.43 - **0.57**
F010	0.43 - **0.57**	0.40 - **0.60**	0.42 - **0.58**
F011	0.35 - **0.65**	0.44 - **0.56**	0.36 - **0.64**
F012	0.49 - 0.51	**0.54** - 0.46	0.48 - 0.52
F013	0.37 - **0.63**	**0.56** - 0.44	0.44 - **0.56**

TABLE IV
RESULTS OF MOS EVALUATION ON SYNTHETIC SPEECH NATURALNESS AND DMOS EVALUATION ON SYNTHETIC SPEECH SPEAKER SIMILARITY WITH 95% CONFIDENCE INTERVALS. WE USED 13 UNSEEN SPEAKERS IN THIS EVALUATION

	d-vec.	Prop. (vec)	Prop. (mat)	Prop. (graph)
MOS	2.8±0.13	**3.1±0.14**	**3.1±0.13**	**3.1±0.14**
DMOS	2.8±0.14	**3.1±0.14**	2.9±0.14	**3.0±0.14**

Tables II and III list the preference scores on the synthetic speech naturalness and speaker similarity, respectively. The bold values denote that there is a significant difference between the two scores ($p < 0.05$). We found that "Prop. (vec)" and "Prop. (graph)" always achieved higher scores than "d-vec." regarding both the naturalness and speaker similarity, which indicated that the proposed similarity-aware speaker embeddings improved the synthetic speech quality in the speaker adaptation task. We observed that "Prop. (mat)" also improved the naturalness; however, it significantly degraded the speaker similarity in a number of cases (e.g., "F005" and "F012").

We further compared the four algorithms on the basis of a five-point scale mean opinion score (MOS) test on the synthetic speech naturalness (1: "very unnatural"; 5: "very natural") and a differential MOS (DMOS) test on the synthetic speech speaker similarity (1: "very dissimilar"; 5: "very similar"). Fifty listeners participated in each of the following evaluations by using our crowdsourced evaluation system. Each listener evaluated 20 speech samples randomly extracted from the 650 (50 × 13) utterances, enabling us to compare average performances of the four algorithms. The total number of task sets was 2 (MOS or DMOS) × 50 (listeners per task set) = 100. Table IV shows the MOS and DMOS evaluation results. Bold values indicate that the method's score was significantly higher than that of "d-vec"

TABLE V

RESULTS OF MOS EVALUATION ON SYNTHETIC SPEECH NATURALNESS AND DMOS EVALUATION ON SYNTHETIC SPEECH SPEAKER SIMILARITY WITH 95% CONFIDENCE INTERVALS. WE USED FIVE SEEN SPEAKERS IN THIS EVALUATION

	d-vec.	Prop. (vec)	Prop. (mat)	Prop. (graph)
MOS	3.1±0.14	3.2±0.14	3.2±0.14	3.1±0.14
DMOS	3.0±0.13	3.1±0.13	2.9±0.12	3.0±0.13

($p < 0.05$). We found that each of the three proposed algorithms significantly improved the synthetic speech naturalness over that of the conventional d-vector. Furthermore, the "Prop. (vec)" and "Prop. (graph)" algorithms also achieved significantly higher DMOS values than with "d-vec." while "Prop. (mat)" did not. These results correspond with the preference AB/XAB test results listed in Tables II and III.

We also conducted subjective evaluations of the four algorithms by using richer DNN architectures than the Feed-Forward networks; these results are described in Appendices C (recurrent speaker encoder) and D (convolutional VAEs).

4) Subjective Evaluation Using Seen Speakers: We also compared the four algorithms in terms of the synthetic speech quality of seen speakers through a series of MOS and DMOS tests. We used five seen speakers labeled as "FP01"–"FP05" in the JNAS corpus. Fifty listeners participated in each of the following evaluations via our crowdsourced evaluation system. Each listener evaluated 20 speech samples randomly extracted from 80 (16 × 5) utterances. The total number of task sets was 2 (MOS or DMOS) × 50 (listeners per task set) = 100. Table V lists the evaluation results. They show that "d-vec." achieved MOS and DMOS values comparable with those of "Prop. (*)," which indicates that the speaker-discrimination-based embedding worked sufficiently well in synthesizing seen speakers' voices.

5) Subjective Evaluation in Speaker Interpolation: We investigated whether the perceptual-similarity-aware speaker embeddings improve the synthetic speech quality in speaker interpolation [57] that aims 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, i.e., it should not deteriorate the speech quality and should provide a way to control the interpolated voice characteristics intuitively. We evaluated the conventional and proposed speaker embeddings in embedding-manipulation-based speaker interpolation [58], [59] that uses a convex combination of speaker embeddings to interpolate their voice characteristics. Formally, if we have two speaker embeddings $\boldsymbol{d}_{\mathrm{A}}$ and $\boldsymbol{d}_{\mathrm{B}}$, an interpolated speaker embedding is calculated as $\boldsymbol{d}_{\mathrm{AB}} = (1-\alpha)\boldsymbol{d}_{\mathrm{A}} + \alpha\boldsymbol{d}_{\mathrm{B}}$ with an interpolation coefficient $0 \leq \alpha \leq 1$. In the speaker interpolation evaluation, we considered four speaker pairs: 1) the first and second most *dissimilar* pairs ("F033-F134" and "F023-F077"), and 2) the first and second most *similar* ones ("F017-F149" and "F088-F122"), to be mixed with a coefficient $\alpha \in \{0.0, 0.25, 0.5, 0.75, 1.0\}$.

We conducted an MOS test on naturalness of the interpolated speech ($\alpha = 0.5$) with speaker embeddings learned by the four different algorithms. Fifty listeners participated in the MOS evaluations by using our crowdsourced evaluation system. Each listener evaluated 16 samples of the interpolated speech. The total number of task sets was 4 (speaker pairs) × 50 (listeners per task) = 200. Table VI lists the MOS evaluation results. From the results, we found that the three proposed algorithms achieved higher MOS values than the conventional d-vector for all speaker pairs. Among the three proposed methods' results, "Prop. (mat)" and "Prop. (graph)" significantly outperformed "d-vec." in the two evaluations using dissimilar speaker pairs (Table VI(a)), and "Prop. (*)" significantly improved the naturalness of interpolated speech over that of "d-vec." for the "F088-F122" speaker pair (Table VI(b)). These results indicated that the proposed similarity-aware speaker embeddings improved the synthetic speech quality not only in speaker adaptation but also in speaker interpolation.

TABLE VI

RESULTS OF MOS EVALUATION ON INTERPOLATED SPEECH NATURALNESS WITH 95% CONFIDENCE INTERVALS. THE **BOLD** SCORES ARE SIGNIFICANTLY HIGHER THAN THOSE OF D-VEC. ($p < 0.05$)

(a) Dissimilar speaker pairs

	d-vec.	Prop. (vec)	Prop. (mat)	Prop. (graph)
F033-F134	2.9±0.13	3.1±0.13	**3.1±0.13**	**3.2±0.13**
F023-F077	3.1±0.13	3.2±0.13	**3.3±0.13**	**3.3±0.13**

(b) Similar speaker pairs

	d-vec.	Prop. (vec)	Prop. (mat)	Prop. (graph)
F017-F149	3.3±0.15	3.3±0.14	3.3±0.14	3.4±0.14
F088-F122	2.9±0.12	**3.2±0.12**	**3.2±0.12**	**3.1±0.12**

We further conducted a variant of the preference XAB test 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 our crowdsourced evaluation system, and each listener evaluated 20 speech samples. The total number of task sets was 4 (speaker pairs) × 30 (listeners per task) = 120. Fig. 16 shows the preference score curves against the five different interpolation coefficients. We observed that the shapes of the curves significantly changed depending on the speaker pair, which 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 speech becomes. To illustrate this observation clearly, we added red lines to Fig. 16 representing "interpolation coefficients and preference XAB scores are equal," i.e., listeners could infer the two speakers' mixing ratio from the interpolated speech perfectly. We can see from Fig. 16(1b) that all the preference scores are near 0.5 regardless of the interpolation coefficient settings, i.e., listeners hardly perceived the differences among the interpolated speech samples. This was a natural result because we mixed very similar speakers' voices, and listeners therefore could not detect the difference between the two speech samples "A" ($\alpha = 0.0$) and "B" ($\alpha = 1.0$). A similar tendency is observed in Fig. 16(2b). However, the score curve for "d-vec." becomes close to the red line, which indicates that the conventional speaker-classification-based embedding space never considered

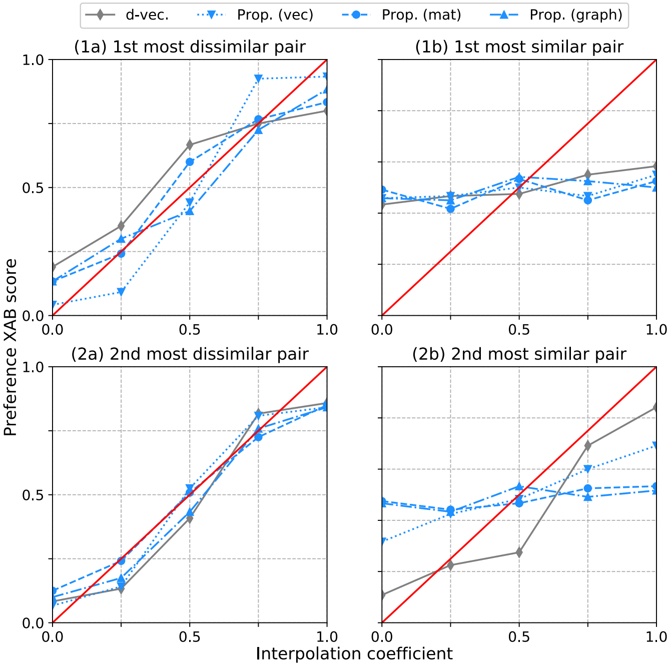


Fig. 16. Results of XAB tests on speaker similarity of the interpolated speech using (1a) the most dissimilar (“F033-F134”), (1b) most similar (“F017-F149”), second most dissimilar (“F023-F077”), and second most similar (“F088-F122”) speaker pairs. The closer the score curves to red lines become, the better listeners inferred two speakers’ mixing ratio from interpolated speech.

TABLE VII
LEAST SQUARES ERRORS BETWEEN THE RED LINES AND CURVES IN FIGS 16(1A) AND (2A)

	d-vec.	Prop. (vec)	Prop. (mat)	Prop. (graph)
F033-F134 (1a)	1.1e−1	6.5e−2	5.6e−2	4.3e−2
F023-F077 (2a)	5.4e−2	4.5e−2	3.9e−2	4.5e−2

the perceptual similarity among speakers in the speaker interpolation. Focusing on the results shown in Fig. 16(1a) and (2a), one can quantify controllability of speaker interpolation using a pair of dissimilar speakers as the distance between the red lines and the curve in this figure. We calculated the linear least squares error between the red lines and preference scores we obtained. Table VII lists the calculation results. They show that the three proposed algorithms decreased the errors better than “d-vec.” did, although the improvement was not significant. These results suggest that the consideration of perceptual similarity among speakers in deep speaker representation learning can improve not only the quality but also the controllability of synthetic speech in multi-speaker generative modeling.

D. Evaluation of Active Learning

We investigated the active learning’s effectiveness in each of the three proposed representation learning algorithms independently. In addition to the three query strategies described in Section III-D2 (“LSF,” “HSF,” and “MSF”), we compared “FS” and “PS (50%)” that trained a speaker encoder with 115 epochs using the fully observed scores and partially observed ones as shown in Fig. 11(a) and (b), respectively.

TABLE VIII
MOS RESULTS OF PROPOSED ALGORITHMS USING ACTIVE LEARNING. THE SECOND COLUMN DENOTES PERCENTAGES OF THE NUMBER OF SCORED SPEAKER PAIRS. **BOLD** SCORES ARE COMPARABLE TO THOSE OF “FS” ($p > 0.05$)

		Prop. (vec)	Prop. (mat)	Prop. (graph)
PS	50.0%	2.9±0.14	**3.0±0.13**	2.9±0.14
MSF	62.5%	3.0±0.12	**3.0±0.13**	**3.1±0.13**
	75.0%	**3.1±0.13**	**3.0±0.13**	**3.1±0.14**
	87.5%	**3.2±0.13**	**3.0±0.13**	**3.1±0.14**
FS	100%	3.2±0.13	3.0±0.12	3.2±0.13

TABLE IX
DMOS RESULTS OF PROPOSED ALGORITHMS USING ACTIVE LEARNING. THE SECOND COLUMN DENOTES PERCENTAGES OF THE NUMBER OF SCORED SPEAKER PAIRS. **BOLD** SCORES ARE COMPARABLE TO “FS” ($p > 0.05$)

		Prop. (vec)	Prop. (mat)	Prop. (graph)
PS	50.0%	2.9±0.14	**2.9±0.13**	2.9±0.13
MSF	62.5%	**3.0±0.14**	**2.9±0.13**	**3.0±0.13**
	75.0%	**3.0±0.14**	**3.0±0.13**	**3.0±0.13**
	87.5%	**3.1±0.14**	**3.0±0.13**	**3.1±0.13**
FS	100%	3.1±0.14	3.0±0.13	3.1±0.14

1) *Evaluation of AUC Improvement:* We investigated how the proposed active learning affected the AUC of similar speaker-pair detection. Fig. 17 shows the curves of the AUC against the active learning iterations. Red and blue lines denote the final AUC values of “FS” and “PS (50%)” after 115 epochs, respectively. Note that the final AUC values of “LSF,” “HSF,” and “MSF,” in Fig. 17(a) did not necessarily correspond to those of “FS” because their speaker encoders sequentially learned perceptual similarity among the 140 seen speakers using differently ordered similarity scores. We found that the query strategies significantly affected the AUC improvement by the active learning and “MSF” reasonably worked among the three strategies, regardless of the training algorithms we used. We observed that active learning in “Prop. (vec)” and “Prop. (graph)” successfully improved the AUC through the iterations better than “PS (50%)” did in both the “Seen-Seen” and “Seen-Unseen” speaker-pair cases. Meanwhile, “Prop. (mat)” increased the AUC by the active learning iterations in the “Seen-Seen” speaker-pair case but resulted in decreasing the AUC in the “Seen-Unseen” case. This result indicates that this algorithm tends to be highly sensitive to the data we use during the proposed active learning.

2) *Evaluation of Synthetic Speech Quality:* We investigated whether the active learning efficiently trained a speaker encoder that improved synthetic speech quality with fewer number of scoring times and training iterations. Similar to Section IV-C3, we conducted MOS and DMOS tests comparing the quality of synthetic speech made by speaker embeddings of “FS,” “PS,” and “MSF” with three different active learning iterations to increase the percentage of scored speaker pairs, to 62.5% (30 iterations), 75% (60 iterations), and 87.5% (90 iterations), respectively. Tables VIII and IX list the results of the MOS and DMOS tests, respectively. We found that “MSF” achieved a synthetic speech quality comparable to that of “FS” with a fewer number of additional similarity score observations and active learning iterations. These results demonstrated that active learning of the perceptual-similarity-aware speaker embeddings effectively reduced the number of scoring times while achieving

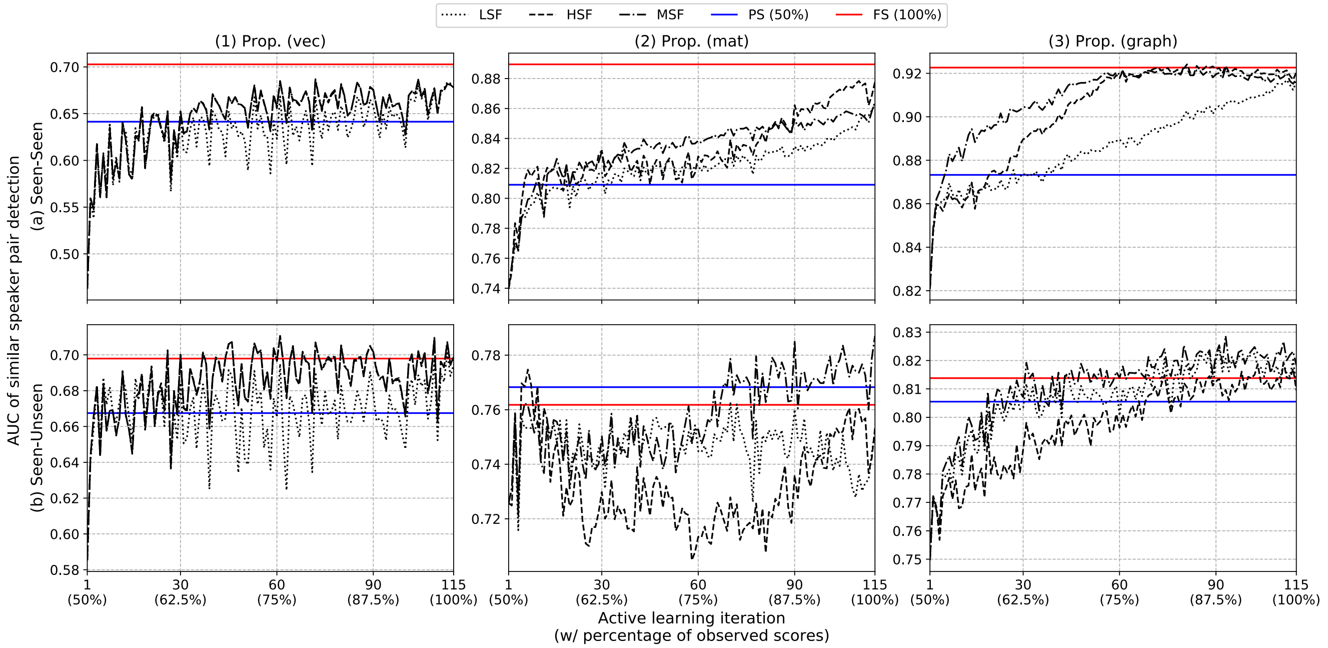


Fig. 17. Curves of similar speaker-pair detection AUC with respect to the number of active learning iterations. We started this active learning with 50% of the observed similarity scores.

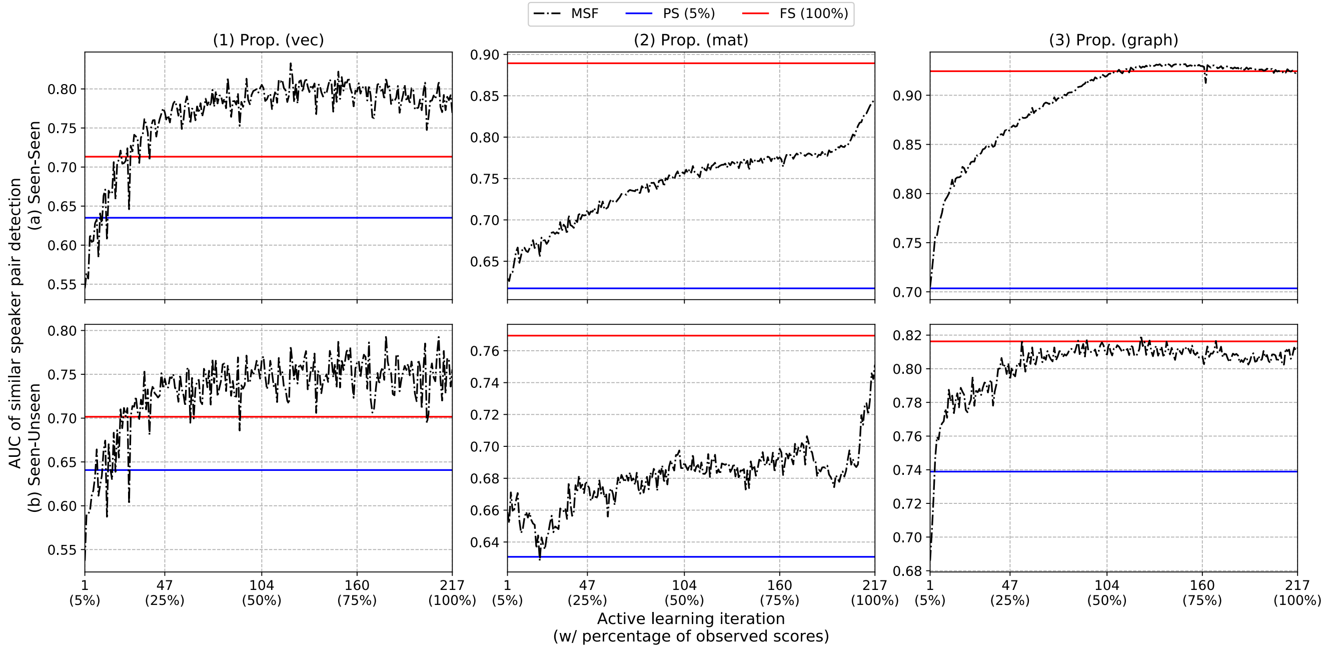


Fig. 18. Curves of similar speaker-pair detection AUC with respect to the number of active learning iterations. We started this active learning with 5% of the observed similarity scores.

higher synthetic speech quality with fewer training iterations. Focusing on the results of "Prop. (mat)," there were no significant differences among the five scores, and "FS" marked the lowest scores in the same row entries.

3) Evaluation of Active Learning Starting With More Limited Observed Scores: We also evaluated the proposed active learning algorithm using the best query strategy, i.e., "MSF," in a more challenging experimental setting. Here, we first divided the 140 seen speakers into 20 groups including seven disjoint speakers (i.e., "F014"–"F020," "F021"–"F026," etc.), and we assumed that speaker similarity scores were observed only among the speakers within each group. Therefore, the percentage of initially observed similarity scores was ${}_{7}\mathrm{C}_{2}/{}_{140}\mathrm{C}_{2} \times 20 = 420/9730 < 5\%$. We then started the proposed active learning with less than 5% of the observed similarity scores and ran 217 epochs to reach the point of 100% score observation.

Fig. 18 shows curves of the AUC with respect to the number of active learning iterations. The red and blue lines denote the final AUC values for "FS" and "PS (5%)," respectively, after 217 epochs. The results show that the active learning in "Prop. (vec)" and "Prop. (graph)" worked well, similarly to the case of starting with 50% of the observed similarity scores (Fig. 17). Notably, "MSF" even outperformed "FS," as shown in Fig. 18(1). This result suggests that "FS" is not necessarily an optimal setting to learn perceptual-similarity-aware speaker embeddings for "Prop. (vec)," and that active learning with the "MSF" query strategy can give better speaker embeddings to capture the perceptual similarity well. "Prop. (graph)" also achieved AUC values comparable with those of "FS" through the active learning iterations, while "Prop. (mat)" did not. This result indicates that "Prop. (mat)" strongly depends on the similarity score observations during active learning.

V. Conclusion

This paper 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 from large-scale perceptual scoring as the target for the speaker encoder training. The algorithms learn speaker embeddings with three different representations of the matrix: a set of vectors, the Gram matrix, and a graph. To reduce costs of scoring and training, we further proposed an active learning algorithm that iterates the perceptual similarity scoring and speaker encoder training. The algorithm selects speaker pairs to be scored next on the basis of the sequentially-trained speaker encoder's similarity prediction results. The experimental evaluation results demonstrated that 1) the proposed speaker representation learning algorithms learned speaker embeddings strongly correlated with perceptual similarity scores, 2) the embeddings improved synthetic speech quality in speech autoencoding tasks better than conventional d-vectors obtained by discriminative modeling, 3) the proposed active learning algorithm achieved higher synthetic speech quality while reducing costs of scoring and training, and 4) among the proposed similarity {vector, matrix, graph} embedding algorithms, the first achieved the best speaker similarity for synthetic speech, and the third gave the highest AUC value for similar speaker-pair detection and the most improvement in the synthetic speech naturalness.

In the future, we will investigate different parameterization of the similarity scores (e.g., using the interval $[0, 1]$, where 1 means "similar" while 0 means "dissimilar") for the perceptual similarity scoring and the hyperparameter settings of the proposed active learning. We will also examine the effect of averaging the speaker embedding during training.

Appendix A
Perceptual Speaker Similarity Scoring Including Same-Speaker Pairs

We present the results of perceptual similarity scoring among the 13 unseen speakers ("F001"–"F013") including both same-speaker and different-speaker pairs. We scored the perceptual similarity of 91 speaker pairs including the 13 same-speaker pairs and ${}_{13}\mathrm{C}_2$ different-speaker ones. The scoring procedure was similar to the one described in Section IV-A1.

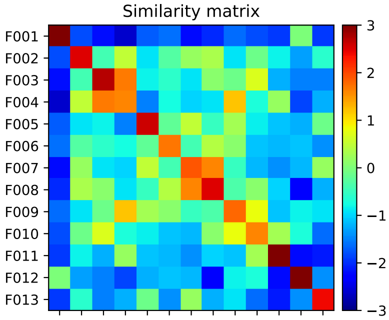


Fig. 19. Perceptual speaker similarity matrix obtained by large-scale perceptual scoring for 13 female Japanese speakers. The listeners scored pairs of both the same speaker and different speakers.

Fig. 19 shows the similarity matrix that we obtained. We found that off-diagonal elements of the matrix became larger than those shown in Fig. 6(b), which indicates that perceptual similarity scoring including the same-speaker pairs tends to increase the similarity score results for different speaker pairs. Regarding the scoring results for the same-speaker pairs' perceptual similarity, the mean and standard deviation of the diagonal elements in the similarity matrix were 2.33 and 0.52, respectively. This result suggests that the crowdsourced listeners did not always score the same-speaker pairs with the maximum value. Hence, we should consider this tendency when extending the proposed algorithms to ones that can model not only the inter-speaker similarity but also the intra-speaker similarity. We also created a histogram of the speaker-pairwise scores of the 13 unseen speakers, as shown in Fig. 20. We observed that the score distributions had larger variances than those shown in Fig. 13. These results indicate that perceptual similarity scoring including same-speaker pairs is more difficult than that using only different-speaker pairs.

Appendix B
Comparison of Predicted and Sampled VAE Latent Variables

In Section IV-C3, we generated the target speaker's acoustic features by using VAE latent variables predicted from the speaker's acoustic features themselves. However, such latent variables might contain the speaker's individual voice characteristics such as his/her speaking style. To remove the target speaker leakage in the VAE latent variables, we also used latent variables sampled from their prior distribution, i.e., $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$, for acoustic feature generation. In this evaluation, we fed a joint vector of the sampled latent variables, PPGs, and speaker embeddings into the VAE decoder networks and generated the target speaker's static-dynamic mel-cepstral coefficients. We conducted a series of preference AB/XAB tests to compare the quality of synthetic speech generated by using the predicted or sampled latent variables. Fifty listeners participated in each of the following evaluations via our crowdsourced evaluation system. Each listener evaluated 10 speech samples randomly extracted from 650 (50 × 13) utterances, enabling us to investigate the average performance in each evaluation case. The total

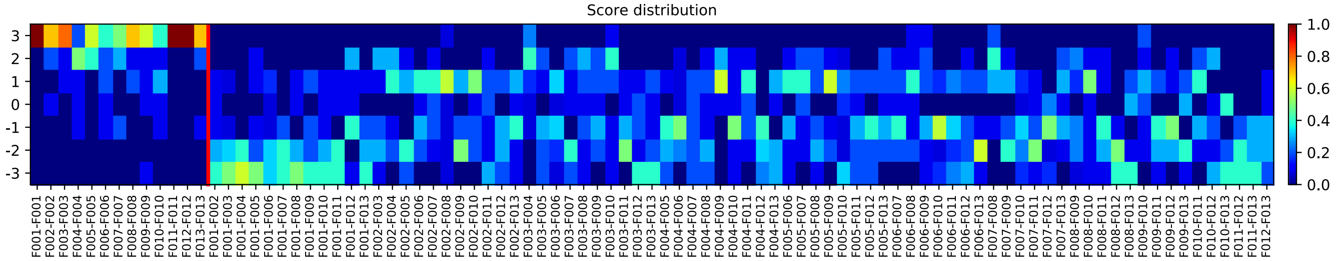


Fig. 20. Histogram of the perceptual similarity scores of the 13 speakers (from "F001" to "F013"). The listeners scored pairs of both the same speaker and different speakers. The first through thirteenth columns represent the perceptual similarity score distributions of the same-speaker pairs.

TABLE X
Preference Scores for Synthetic Speech Naturalness (Left: Predicted VAE Latent Variables; Right: Sampled Ones)

	Predicted	vs.	Sampled
d-vec.	0.47	-	0.53
Prop. (vec)	0.52	-	0.48
Prop. (mat)	0.49	-	0.51
Prop. (graph)	0.52	-	0.48

TABLE XI
Preference Scores for Synthetic Speech Speaker Similarity (Left: Predicted VAE Latent Variables; Right: Sampled Ones)

	Predicted	vs.	Sampled
d-vec.	0.52	-	0.48
Prop. (vec)	0.52	-	0.48
Prop. (mat)	0.50	-	0.50
Prop. (graph)	0.50	-	0.50

number of task sets was 2 (AB or XAB) × 50 (listeners per task set) × 4 (algorithms) = 400.

Tables X and XI list the preference scores for the synthetic speech naturalness and speaker similarity, respectively. From these tables, we found no significant differences between the "Predicted" and "Sampled" scores. These results indicate that target speaker leakage did not occur in the VAE-based multi-speaker generative modeling; therefore, experimental evaluation using the predicted VAE latent variables is sufficient to demonstrate the proposed algorithms' effectiveness.

Appendix C
Subjective Evaluation Using Strong D-Vector Baseline

In Section IV-C3, we used simple Feed-Forward DNNs for the conventional algorithm. Here, we instead adopted a three-layer uni-directional long-short term memory (LSTM)-based speaker encoder ("d-vec. (LSTM)") [60] and investigated the effectiveness of the strong baseline in improving the synthetic speech quality. The speaker encoder had 256 memory cells.

We first compared the two baseline methods, "d-vec." and "d-vec. (LSTM)," and evaluated only the recurrent DNN architecture's effectiveness. Fifty listeners participated in this evaluation through a series of preference AB/XAB tests via our crowdsourced evaluation system. Each listener evaluated 10 speech samples randomly extracted from 650 (50 × 13) utterances. The total number of task sets was 2 (AB or XAB) × 50 (listeners per task set) = 100. Table XII lists the preference scores for the synthetic speech naturalness and speaker similarity. From the results, we found that "d-vec. (LSTM)" significantly outperformed "d-vec." in terms of both the naturalness and the speaker similarity of synthetic speech. These results indicate that the recurrent speaker encoder was effective for improving the synthetic speech quality in multi-speaker acoustic modeling.

TABLE XII
Preference Scores for Synthetic Speech Naturalness and Speaker Similarity (Left: Feed-Forward DNN; Right: LSTM)

	d-vec.	vs.	d-vec. (LSTM)
Naturalness	0.32	-	**0.68**
Similarity	0.37	-	**0.63**

TABLE XIII
Results of MOS Evaluation on Synthetic Speech Naturalness and DMOS Evaluation on Synthetic Speech Speaker Similarity With 95% Confidence Intervals. We Used an LSTM-Based Speaker Encoder for "d-vec."

	d-vec. (LSTM)	Prop. (vec)	Prop. (mat)	Prop. (graph)
MOS	3.1±0.13	3.0±0.14	3.0±0.13	3.0±0.13
DMOS	2.9±0.14	**3.1±0.15**	3.0±0.14	3.0±0.13

We then compared "d-vec. (LSTM)" with "Prop. (*)" through a series of MOS and DMOS tests and investigated whether the three proposed algorithms could even outperform the strong baseline in terms of the synthetic speech quality. Fifty listeners participated in this evaluation via our crowdsourced evaluation system. Each listener evaluated 20 speech samples randomly extracted from 650 (50 × 13) utterances. The total number of task sets was 2 (MOS or DMOS) × 50 (listeners per task set) = 100. Table XIII lists the MOS and DMOS evaluation results. From the MOS results, "d-vec. (LSTM)" achieved naturalness comparable with those of "Prop. (*)" because of the temporal dependency modeling by the LSTM-based speaker encoder. However, "Prop. (vec)" significantly outperformed "d-vec. (LSTM)" in terms of the speaker similarity of synthetic speech, although we did not use the LSTM-based speaker encoder for the proposed algorithms. This result indicate the algorithm's effectiveness in improving the synthetic speech quality.

Appendix D
Subjective Evaluation Using Rich DNN Architecture for VAEs

In Section IV-C3, we used simple Feed-Forward DNNs for the VAE-based multi-speaker acoustic model. Here, we adopted

TABLE XIV
RESULTS OF MOS EVALUATION ON SYNTHETIC SPEECH NATURALNESS AND DMOS EVALUATION ON SYNTHETIC SPEECH SPEAKER SIMILARITY WITH 95% CONFIDENCE INTERVALS. WE USED GATED-CNN-BASED VAES IN THIS EVALUATION

	d-vec.	Prop. (vec)	Prop. (mat)	Prop. (graph)
MOS	3.1±0.13	3.2±0.14	3.2±0.14	**3.3±0.15**
DMOS	3.0±0.14	**3.2±0.15**	3.1±0.14	3.1±0.14

gated convolutional neural networks (CNNs) [61] as the DNN architecture for the acoustic model and investigated this richer architecture's effectiveness in improving the synthetic speech quality. The CNN-based encoder and decoder had two 1D convolutional (Conv1D) layers and two 1D deconvolutional (Deconv1D) layers, respectively, along the temporal axis. We set the convolution window size and stride width of all the Conv1D and Deconv1D layers to five and one, respectively. The encoder extracted the framewise 64-dimensional latent variables from a joint vector of the static mel-cepstral coefficients and PPGs. The numbers of input channels for the first and second Conv1D layers were 256 and 128, respectively. The decoder reconstructed the static mel-cepstral coefficients from a joint vector of the latent variables, PPGs, and 8-dimensional speaker embeddings. The numbers of input channels for the first and second Deconv1D layers were 128 and 256, respectively.

We conducted a series of MOS and DMOS tests and compared the four CNN-based VAEs for acoustic modeling with speaker embeddings learned by each of the four algorithms. Fifty listeners participated in this evaluation via our crowdsourced evaluation system. Each listener evaluated 20 speech samples randomly extracted from 650 (50×13) utterances. The total number of task sets was 2 (MOS or DMOS) $\times$ 50 (listeners per task set) $=$ 100. Table XIV lists the MOS and DMOS evaluation results. They show that the overall MOS and DMOS values increased in comparison with using the Feed-Forward DNNs as acoustic models (Table IV). Among the four algorithms, "Prop. (graph)" and "Prop. (vec)" achieved the best MOS and DMOS values, which were significantly better than those of "d-vec." baseline. These results demonstrate that these algorithms were also effective in multi-speaker acoustic modeling using the CNN-based VAEs.

REFERENCES

[1] S. Latif, R. Rana, S. Khalifa, R. Jurdak, J. Qadir, and Björn W. Schuller, "Deep representation learning in speech processing: challenges, recent advances, and future trends," 2020, *arXiv:2001.00378*.

[2] T. Kinnunen and H. Li, "An overview of text-independent speaker recognition: From features to supervectors," *Speech Commun.*, vol. 52, no. 1, pp. 12–40, Dec. 2010.

[3] F. Bimbot *et al.*, "A tutorial on text-independent speaker verification," *EURASIP J. Appl. Signal Process.*, vol. 4, pp. 430–451, Apr. 2004.

[4] 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. Int. Conf. Acoust., Speech, Signal Process.*, Florence, Italy, May 2014, pp. 4080–4084.

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

[6] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 4, pp. 788–798, May 2011.

[7] Y. Sagisaka, "Speech synthesis by rule using an optimal selection of non-uniform synthesis units," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, New York, USA, Apr. 1988, pp. 679–682.

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

[9] Y. Stylianou, O. Cappé, and E. Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Trans. Speech Audio Process.*, vol. 6, no. 2, pp. 131–142, Mar. 1988.

[10] T. Toda, A. W. Black, and K. Tokuda, "Voice conversion based on maximum likelihood estimation of spectral parameter trajectory," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 8, pp. 2222–2235, Nov. 2007.

[11] A. Oord *et al.*, "WaveNet: A generative model for raw audio," in *Proc. SSW*, Sunnyvale, USA, Sep. 2016, p. 125.

[12] 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.

[13] Y. Wang, RJ *et al.*, "Tacotron: Towards end-to-end speech synthesis," in *Proc. INTERSPEECH*, Stockholm, Sweden, Aug. 2017, pp. 4006–4010.

[14] Y. Ren *et al.*, "FastSpeech: Fast, robust and controllable text to speech," in *Proc. NeurIPS*, Vancouver, Canada, Dec. 2019, pp. 3171–3180.

[15] Y. Ren *et al.*, "FastSpeech 2: Fast and high-quality end-to-end text to speech," in *Proc. Int. Conf. Learn. Representations (ICLR)*, Vienna, Austria, May 2021.

[16] S. Vasquez and M. Lewis, "MelNet: A. generative model for audio in the frequency domain," 2019, *arXiv:1906.01083*.

[17] Y. Saito, S. Takamichi, and H. Saruwatari, "Statistical parametric speech synthesis incorporating generative adversarial networks," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 26, no. 1, pp. 84–96, Jan. 2018.

[18] R. Yamamoto, E. Song, and J. Kim, "Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Barcelona, Spain, May 2020, pp. 6199–6203.

[19] R. Doddipatla, N. Braunschweiler, and R. Maia, "Speaker adaptation in DNN-based speech synthesis using d-vectors," in *Proc. INTERSPEECH*, Stockholm, Sweden, Aug. 2017, pp. 3404–3408.

[20] Y. Saito, Y. Ijima, K. Nishida, and S. Takamichi, "Non-parallel voice conversion using variational autoencoders conditioned by phonetic posteriorgrams and d-vectors," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Alberta, Canada, Apr. 2018, pp. 5274–5278.

[21] Y. Jia *et al.*, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," in *Proc. NeurIPS*, Montreal, Canada, Dec. 2018, pp. 4480–4490.

[22] F. Fang *et al.*, "Speaker anonymization using x-vector and neural waveform models," in *Proc. SSW*, Vienna, Austria, Sep. 2019, pp. 155–160.

[23] 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. Int. Conf. Acoust., Speech, Signal Process.*, Brighton, U.K., May 2019, pp. 6161–6165.

[24] J. González, Z. Dai, A. C. Damianou, and N. D. Lawrence, "Preferential bayesian optimization," in *Proc. Int. Conf. Mach. Learn.*, Sydney, Australia, Aug. 2017, pp. 1282–1291.

[25] Y. Koyama, I. Sato, and M. Goto, "Sequential gallery for interactive visual design optimization," *ACM Trans. Graph.*, vol. 39, no. 4, pp. 88:1–88:12, Jul. 2020.

[26] J. Yamagishi, T. Kobayashi, Y. Nakano, K. Ogata, and J. Isogai, "Analysis of speaker adaptation algorithms for HMM-based speech synthesis and a constrained SMAPLR adaptation algorithm," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 17, no. 1, pp. 66–83, Jan. 2009.

[27] Z. Wu, P. Swietojanski, C. Veaux, S. Renals, and S. King, "A study of speaker adaptation for DNN-based speech synthesis," in *Proc. INTERSPEECH*, Dresden, Germany, Sep. 2015, pp. 879–883.

[28] E. Cooper *et al.*, "Zero-shot multi-speaker text-to-speech with state-of-the-art neural speaker embeddings," in *Proc. Int. Conf. Acoust., Speech Signal Process.*, Barcelona, Spain, May 2020, pp. 6184–6188.

[29] Y. Saito, S. Takamichi, and H. Saruwatari, "DNN-based speaker embedding using subjective inter-speaker similarity for multi-speaker modeling in speech synthesis," in *Proc. SSW*, Vienna, Austria, Sep. 2019, pp. 51–56.

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

[31] B. Settles, "Active learning literature survey," Dept. Comput. Sci., Univ. Winconsin-Madison, Tech. Rep. 1648, Jan. 2009.

[32] H.-T. Luong, S. Takaki, G. E. Henter, and J. Yamagishi, "Adapting and controlling DNN-based speech synthesis using input codes," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, New Orleans, USA, May 2017, pp. 1905–1909.

[33] N. Hojo, Y. Ijima, and H. Mizuno, "DNN-based speech synthesis using speaker codes," *IEICE Trans. Inf. Syst.*, vol. E101-D, no. 2, pp. 462–472, Feb. 2018.

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

[35] M. Tachibana, T. Nose, J. Yamagishi, and T. Kobayashi, "A technique for controlling voice quality of synthetic speech using multiple regression HSMM," in *Proc. Int. Conf. Spoken Lang. Process.*, Pittsburgh, USA, Sep. 2006, pp. 2438–2441.

[36] 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. Int. Conf. Spoken Lang. Process.*, Bonn, Germany, Aug. 2007, pp. 101–106.

[37] 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 Commun.*, vol. 99, pp. 135–143, May 2018.

[38] Z. Liu and B. Mak, "Multi-lingual multi-speaker text-to-speech synthesis for voice cloning with online speaker enrollment," in *Proc. INTERSPEECH*, Shanghai, China, 2020, pp. 2932–2936.

[39] L. Zhang *et al.*, "DurIAN-SC: Duration informed attention network based singing voice conversion system," in *Proc. INTERSPEECH*, Shanghai, China, 2020, pp. 1231–1235.

[40] J. R. Hershey, Z. Chen, J. L. Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Shanghai, China, Mar. 2016, pp. 31–35.

[41] 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 Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.

[42] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.

[43] F. M. Zanzotto, "Viewpoint: Human-in-the-loop artificial intelligence," *J. Artif. Intell. Res.*, vol. 64, pp. 243–252, Feb. 2019.

[44] K. Fujii, Y. Saito, S. Takamichi, Y. Baba, and H. Saruwatari, "HumanGAN: Generative adversarial network with human-based discriminator and its evaluation in speech perception modeling," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Barcelona, Spain, May 2020, pp. 6239–6243.

[45] K. Itou *et al.*, "JNAS: Japanese speech corpus for large vocabulary continuous speech recognition research," *J. Acoustical Soc. Jpn. (E)*, vol. 20, no. 3, pp. 199–206, May 1999.

[46] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Istanbul, Turkey, Jun. 2000, pp. 1315–1318.

[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 Commun.*, vol. 27, no. 3-4, pp. 187–207, Apr. 1999.

[48] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, Jul. 2011.

[49] 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. Int. Conf. Multimedia Expo.*, Seattle, USA, Jul. 2016, pp. 1–6.

[50] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. Int. Conf. Artif. Intell. Statist.*, Lauderdale, USA, Apr. 2011, pp. 315–323.

[51] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2013, *arXiv:1312.6114*.

[52] 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. Second Int. Workshop Models Anal. Vocal Emissions Biomed. Appl.*, Florence, Italy, Sep. 2001, pp. 1–6.

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

[54] J. L. Fleiss, "Measuring nominal scale agreement among many raters," *Psychol. Bull.*, vol. 76, no. 5, pp. 378–382, 1971.

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

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

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

[58] 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.

[59] 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.

[60] Q. Wang, C. Downey, L. Wan, P. A. Mansfield, and I. L. Moreno, "Speaker diarization with LSTM," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Alberta, Canada, Apr. 2018, pp. 5239–5243.

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

Yuki Saito (Student Member, IEEE) received the M.S. degree in 2018 from the Graduate School of Information Science and Technology, The University of Tokyo, Tokyo, Japan, where he is currently working toward the Ph.D. degree. His research interests include speech synthesis, voice conversion, and machine learning. He was the recipient of eight paper awards including the 34th TELECOM System Technology Award for Students from the Telecommunications Advancement Foundation. He is a Student Member of the Acoustical Society of Japan.

Shinnosuke Takamichi (Member, IEEE) received the Ph.D. degree from the Graduate School of Information Science, Nara Institute of Science and Technology, Ikoma, Japan, in 2016. He is currently an Assistant Professor with The University of Tokyo, Tokyo, Japan. He was the recipient of more than ten paper or achievement awards including the 3rd IEEE SPS Japan Young Author Best Paper Award.

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