ZERO-SHOT MULTI-SPEAKER TEXT-TO-SPEECH WITH STATE-OF-THE-ART NEURAL SPEAKER EMBEDDINGS

Erica Cooper[⋆], *Cheng-I Lai*[†], *Yusuke Yasuda*[⋆], *Fuming Fang*[⋆], *Xin Wang*[⋆], *Nanxin Chen*[‡], *Junichi Yamagishi*[⋆]

[⋆] National Institute of Informatics, Tokyo, Japan
[†] Massachusetts Institute of Technology, Cambridge, USA [‡] Johns Hopkins University, Baltimore, USA

ABSTRACT

While speaker adaptation for end-to-end speech synthesis using speaker embeddings can produce good speaker similarity for speakers seen during training, there remains a gap for zero-shot adaptation to unseen speakers. We investigate multi-speaker modeling for end-to-end text-to-speech synthesis and study the effects of different types of state-of-the-art neural speaker embeddings on speaker similarity for unseen speakers. Learnable dictionary encoding-based speaker embeddings with angular softmax loss can improve equal error rates over x-vectors in a speaker verification task; these embeddings also improve speaker similarity and naturalness for unseen speakers when used for zero-shot adaptation to new speakers in end-to-end speech synthesis.

Index Terms— Speech synthesis, speaker adaptation, speaker embeddings, transfer learning, speaker verification

1. INTRODUCTION

Recent advances in end-to-end text-to-speech (TTS) synthesis have enabled us to produce very realistic and natural-sounding synthetic speech [1, 2] with mean opinion scores (MOS) approaching those of natural human speech [3]. Not only speaker dependent TTS systems but also multi-speaker TTS systems show remarkable results [4]. However, adapting voice models to arbitrary new speakers using a small amount of data (speaker adaptation) remains a challenge.

An effective approach for speaker adaptation in neural TTS is to fine-tune all or part of model with a small amount of data from the target speaker [5, 6]. This approach can also be used to adapt to new speaking styles such as Lombard speech [7]. A different but complementary approach is to use speaker embeddings to model speaker identity in TTS. Prior studies have focused on training a speaker encoder network jointly with the TTS model [6, 8, 9] or the neural vocoder [10]; others have explored the use of speaker embeddings in combination with fine-tuning the TTS model [10, 11, 12]. Approaches that use fine-tuning necessarily require transcribed adaptation data, as well as more computational time and resources to adapt to a new speaker. Furthermore, speaker encoder networks that are jointly trained with the TTS model cannot benefit from data outside of the TTS training data, which is restricted to be of relatively high quality in clean recording conditions.

Transfer learning for speaker modeling in TTS addresses these issues. With this approach, the speaker embedding network is trained completely separately, perhaps for a different task such as speaker recognition. The benefit of this approach is that speaker recognition models can be trained on a large amount of data that does not have to be of the same high quality typically required for TTS, and these models can obtain robust speaker representations that are independent of channel and recording conditions using relatively small amounts of target speaker data, which does not necessarily have to be transcribed. End-to-end synthesis models can then be used to adapt to a target speaker's voice in a zero-shot manner by using the speaker embedding only, without necessarily needing to fine-tune the entire model. Several recent studies [9, 13, 14, 15] have used this approach for speaker modeling in TTS, with [15] modeling both speaker and language characteristics. [13] observed that unseen speakers' synthetic speech had lower speaker similarity to the target speaker than seen speakers, accents were often mismatched, and nuances such as characteristic prosody were lost, indicating that while seen speakers can be well-modeled in this manner, there is room for improvement for modeling unseen speakers.

In parallel with the above-mentioned studies, there has been substantial development in end-to-end speaker recognition. Villalba et al. summarized several state-of-the-art speaker recognition systems for the NIST SRE18 Challenge [16], where x-vector based systems [17] consistently outperformed i-vector based systems [18]. There has also been a surge of interest in new encoding methods and end-to-end loss functions for speaker recognition [19, 20, 21, 22, 23, 24, 25]. One prominent advancement is the use of learnable dictionary encoding (LDE) [19] and angular softmax [20] for speaker recognition, which are reported to boost the speaker recognition performance on open-source corpora such as the VoxCelebs [26, 27].

One aspect of our study is therefore an attempt to find out how effective these recent developments in speaker verification are for speaker adaption in TTS. More specifically we investigate the capability of neural speaker embeddings [16, 17, 19] to capture and model characteristics of speakers that were unseen during TTS model training. For this purpose, we extend an improved Tacotron system in [28] to a multi-speaker TTS system and conduct systematic analysis to answer the above question. We also analyze how the quality and similarity of generated voices are correlated with automatic speaker verification (ASV) accuracy.

While prior studies have focused on transfer learning for zero-shot speaker adaptation for end-to-end TTS, to our knowledge this is the first investigation of many different types of speaker embeddings to determine whether some type of embedding is best for modeling unseen speakers and to learn whether the best embeddings for ASV are the same as the best embeddings for TTS.

2. NEURAL SPEAKER EMBEDDINGS

There are three components in a typical end-to-end speaker recognition system: an encoder network, a statistical pooling layer, and a classifier [16]. An encoder network acts as a frame-level feature extractor, the statistical pooling layer summarizes frame-level representations to a fixed-dimensional utterance-level embedding, and

The second author performed the work mostly while interning at NII.

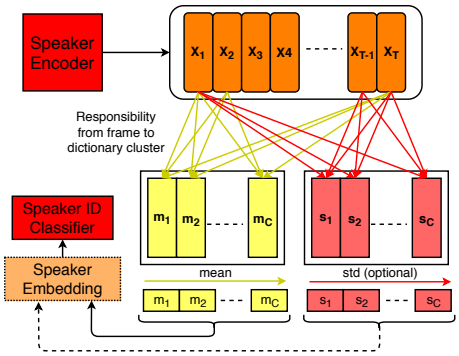

Fig. 1. Learnable dictionary encoding (LDE) pooling method in an end-to-end speaker recognition system.

the classifier determines the speaker identity based on the embedding. In most cases, the neural speaker embedding is obtained after pooling and before classification. Below, we describe each of these components used in our work.

2.1. Encoder Network

In the original x-vector paper [17], a time delay neural network (TDNN) was used as the encoder network and its variants were explored in [16]. The TDNN is composed of 1D convolution and fully connected layers. Several later studies [16, 19, 21, 22, 25, 27] suggest replacing the TDNN with variants of ResNet34, composed of 2D convolutions, as the encoder network. We used TDNN and ResNet34 for x-vector and LDE embeddings, respectively.

2.2. Pooling methods

The pooling method is an important component since it summarizes frame-level representations into a fixed-dimensional utterance-level embedding.

Statistical Pooling (SP): A statistical pooling layer was adopted in the original x-vector paper [17]. It computes the mean and standard deviation of the frame-level representations, which are concatenated as a fixed-dimensional vector.

Learnable Dictionary Encoding (LDE): Instead of the single mean and standard deviation as in SP, the LDE layer proposed in [19] conducts soft clustering of the frame-level representations and concatenates the clusters' means and standard deviations.

Given the frame-level representations $\mathbf{x}_T = \{x_1, x_2, ..., x_T\}$ from the encoder networks, where T is the sequence length, an LDE layer learns a dictionary of C clusters $\{e_1, e_2, ..., e_C\}$. The learning procedure is decomposed into three steps: 1) compute some distance r_{tc} from each frame x_t to each cluster e_c, 2) learn a soft cluster weight w_{tc} of x_t to e_c based on r_{tc}, and 3) aggregate x_t based on w_{tc} over time T to yield an utterance-level representation. Here r_{tc} is L2 distance, $r_{tc} = \|x_t - e_c\|^2$. The cluster weight w_{tc} can be computed by, $w_{tc} = \exp(-r_{tc}) / \sum_{i=1}^{C} \exp(-r_{ti})$. The aggregation of x_t is similar to the supervector notion in [29]. We first compute mean $m_c = \frac{1}{Z} \sum_{\forall t} w_{tc}(x_t - e_c)$ and/or standard deviation $s_c = \frac{1}{Z} \sqrt{\sum_{\forall t} w_{tc}(x_t - e_c)^2}$ for each cluster, which are concatenated for $\forall c \in \{1..C\}$ to form a mean vector $\mathbf{m}_C$, and similarly for a standard deviation vector $\mathbf{s}_C$. Here $Z = \sum_{t=1}^{T} w_{tc}$. Figure 1 illustrates these three steps.

2.3. Classifier

As shown in Figure 1, the last step is to predict speaker IDs via the softmax layer. The standard training criterion is therefore cross entropy. More discriminative criteria called angular softmax loss (A-softmax) and their variants have recently been proposed and evaluated in [20, 24, 30]. The criteria considers angular margins between classes and is expected to produce more separable embedding representations. We examine both normal softmax and angular softmax.

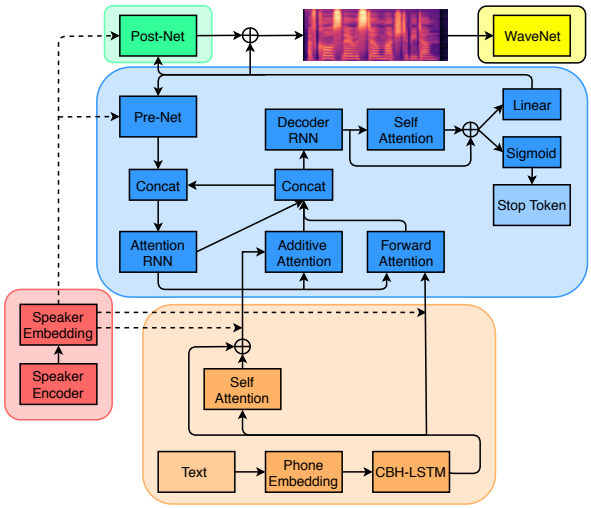

Fig. 2. Proposed multi-speaker TTS system. Encoder blocks are in *orange*, decoder blocks in *blue*, pre-net block in *green*, speaker encoder blocks in *red*, and vocoder block in *yellow*.

3. MULTI-SPEAKER TTS MODEL ARCHITECTURE

The above speaker embedding vectors are used as additional inputs to condition speaker characteristics in our multi-speaker TTS system. Our end-to-end multi-speaker text-to-speech model architecture is based on Tacotron [1], with the extension of self-attention described in [28] to better capture long-range dependencies illustrated in Figure 2. We use phoneme input. We carry out basic rule-based text normalization to expand abbreviations and numbers. We then convert the text to a phoneme representation using flite [31]. A self-attention block [32] is added in the encoder, and so the encoder produces two outputs: one is the original output of the long short-term memory (LSTM), and the other is the output from the self-attention block. The LSTM output is passed to a forward attention block [33], which speeds up the alignment, and the self-attention output is passed to an additive attention block, which allows attention to longer-range information. The outputs of the dual attention mechanism are concatenated before being passed to the decoder. The output of the decoder is an 80-dimensional mel-spectrogram.

We consider three possible locations to input speaker embeddings: concatenating with each of the encoder outputs before inputting to the attention mechanisms, inputting to the prenet to the decoder, or inputting to the postnet. We extract a speaker embedding vector from each training utterance using the speaker encoder and average them per speaker. We then project all speaker embeddings down to 64 dimensions using a dense layer before inputting them to any location in the model.

Speaker adaptation to new speakers is zero-shot. As in the training phase, we extract a speaker embedding vector from each untranscribed adaptation utterance of a target speaker using the speaker encoder. We then input the averaged speaker embedding to generate mel spectrograms of the target speaker. No fine tuning is used. To convert the predicted mel spectrograms into audio, we use a WaveNet [34] vocoder. Input is mel spectrograms and output is

16-bit 16kHz waveforms. The code for our multi-speaker Tacotron implementation and audio samples will be available online[1][2].

4. EXPERIMENTS

4.1. Speaker Verification

In the following sections, we will refer to speaker embeddings based on TDNN+SP as the x-vectors and those based on ResNet34+LDE as the LDEs.

Data: Following [13], we trained speaker verification systems on VoxCeleb1+2 [26, 27]. The training data were all of VoxCeleb2 plus the training portion of VoxCeleb1 (7,325 speakers and 1,277,344 utterances). The clean speech was augmented with reverberation, noise, music, and babble, as described in [17], and then a random subset of these 1,000,000 augmented utterances was combined with the original clean speech. The final training data consisted of 2,277,344 utterances. We report the speaker verification results on the original VoxCeleb1 test set.

Acoustic Features and Pre-processing: We trained x-vectors on 30-dimensional MFCCs and LDEs with 30-dimensional log-Mel filter banks. Kaldi-based 3-second sliding cepstral mean normalization and energy VAD were applied to the acoustic features. This is similar to the setup described in [16]. For the LDE systems, each training sample was a 3-8 second chunk randomly sampled from its original utterance. Our chunk length selection is consistent with [19, 21].

System Details: Our x-vectors are based on the Kaldi recipe[3], with 512 dimensions. For the LDEs, our ResNet34 is the same as [16, 19, 21], and we set the number of dictionary clusters $C = 32$ and mini-batch size to 128. We experimented with the following hyperparameter combinations: embedding dimension $\{512, 256, 200\}$, softmax margin $m \in \{2, 3, 4\}$, and pooling only mean vector $\mathbf{m}_C$ or both mean and standard deviation vectors $\mathbf{m}_C$ and $\mathbf{s}_C$.

Embedding Post-processing and Backend: Our backend is PLDA [35] with score normalization. We followed Kaldi's backend recipe in post-processing the embeddings prior to the PLDA: centering and LDA reduction to 200 dimensions. We also scored the original embeddings without this post-processing step, as we were interested in the effect of such a procedure for speaker adaption in TTS. Note that we did not perform length normalization nor any adaptation/tuning, as normally done in speaker verification, such as [16].

Verification Results: Table 1 presents the results of speaker verification on the VoxCeleb1 test set. We denoted our 7 LDE embeddings as LDE-1, LDE-2, etc, and used superscript a $\mathbf{N}$ to mark those with the post-processing step described above. The LDEs attained results on par with the x-vectors. We also observed that decreasing speaker embedding size, increasing angular margin m, and pooling both $\mathbf{m}_C$ and $\mathbf{s}_C$ improve the performance.

4.2. Preliminary Experiments for Speaker Similarity in TTS

Since the best training method and location for inputting speaker embeddings to the TTS was unknown, we conducted preliminary experiments to learn which settings produce the best speaker similarity for unseen speakers. We wanted to learn whether it was better to train gender-dependent or gender-independent models, and whether it is best to input speaker embeddings to the prenet, concatenate with the encoder output, input at the postnet, or some combination of these.

Data: We used the VCTK corups [37], which consists of read English speech from 109 different speakers in different English dialects. Each speaker read about 400 sentences. Two speakers were excluded due to missing or inadequate data. Four development and four test speakers were held out, chosen to be a mix of genders and dialects, and to have enough unique utterances to have 50 unseen sentences per speaker for TTS evaluation and 50 unseen utterances for "adaptation data" for extracting speaker embeddings. Audio was preprocessed by first high-pass filtering at a cutoff of 80 Hz to remove low-frequency line noise, then normalized using sv56 [38], then trimmed to remove start and end silences. All utterances from the 99 training ("seen") speakers were used to train TTS and to extract speaker embeddings for these speakers; this same data was used to train gender-dependent WaveNet vocoders. The embeddings of the four development and four test speakers ("unseen" speakers) were extracted using only the 50 held-out "adaptation" utterances.

Training: We used a "warm-start" training approach to reduce experimental iteration time. We initialized our multi-speaker models with parameters from a well-trained speaker-dependent model trained on the "Nancy" data from Blizzard 2011 [39] for about 105k steps. We experimentally found that multi-speaker models trained with warm-start for one day (about 40k steps) produced synthetic speech of similar quality to models trained from scratch on VCTK data only for four days. Furthermore, [40] observed that the VCTK corpus has a relatively small number of unique words, whereas the Nancy dataset has more than three times as many; our multi-speaker model can benefit from this increased lexical coverage.

Settings: We tried a number of different settings to determine which had the best similarity for unseen speakers. For speaker embeddings, we used x-vectors. We tried two different training approaches:

- **Gender-independent:** We used data from all VCTK training speakers (male and female) for warm-start training.
- **Gender-dependent:** We ran two separate warm-start trainings, one using only male VCTK training data and the other using only the female data.

At the same time, we tried four different settings for the location to input speaker embeddings:

- Prenet only (`pre`)

Table 1. *Verification results on the original VoxCeleb1 test set.* $\mathbf{m}, \mathbf{s}$ *are mean* $\mathbf{m}_C$ *and standard deviation* $\mathbf{s}_C$ *vectors. S is softmax, AS is A-softmax and the number within parentheses is the angular margin* m*. norm indicates post-processing (centering+LDA) on the embeddings.* $C = 32$ *is set for all LDEs. EER denotes equal error rate and* $\text{DCF}_{0.01}^{min}$ *denotes minimum detection cost function value with a prior value set to 0.01 [36].*

embed.	dim.	pl.	obj.	norm	EER	$\text{DCF}_{0.01}^{min}$
i-Vec$^{\mathbf{N}}$	400	$\mathbf{m}$	EM	✓	5.329	0.493
x-Vec	512	$\mathbf{m}, \mathbf{s}$	S		3.298	0.343
x-Vec$^{\mathbf{N}}$	512	$\mathbf{m}, \mathbf{s}$	S	✓	3.213	0.342
LDE-1	512	$\mathbf{m}$	S		3.415	0.366
LDE-1$^{\mathbf{N}}$	512	$\mathbf{m}$	S	✓	3.446	0.365
LDE-2	512	$\mathbf{m}$	AS(2)		3.674	0.364
LDE-2$^{\mathbf{N}}$	512	$\mathbf{m}$	AS(2)	✓	3.664	0.386
LDE-3	512	$\mathbf{m}$	AS(3)		**3.033**	**0.314**
LDE-3$^{\mathbf{N}}$	512	$\mathbf{m}$	AS(3)	✓	3.171	0.327
LDE-4	512	$\mathbf{m}$	AS(4)		3.112	0.315
LDE-4$^{\mathbf{N}}$	512	$\mathbf{m}$	AS(4)	✓	3.271	0.327
LDE-5	256	$\mathbf{m}$	AS(2)		3.287	0.343
LDE-5$^{\mathbf{N}}$	256	$\mathbf{m}$	AS(2)	✓	3.367	0.351
LDE-6	200	$\mathbf{m}$	AS(2)		3.266	0.396
LDE-6$^{\mathbf{N}}$	200	$\mathbf{m}$	AS(2)	✓	3.266	0.396
LDE-7	512	$\mathbf{m}, \mathbf{s}$	AS(2)		**3.091**	**0.303**
LDE-7$^{\mathbf{N}}$	512	$\mathbf{m}, \mathbf{s}$	AS(2)	✓	3.171	0.328

[1] https://github.com/nii-yamagishilab/multi-speaker-tacotron

[2] https://nii-yamagishilab.github.io/samples-multi-speaker-tacotron

[3] https://github.com/kaldi-asr/kaldi/tree/master/egs/voxceleb/v2

Table 2. *Average cosine similarities between original and synthesized speech from different model configurations for seen (training) and unseen (dev set) speakers. Waveform generation was done using unseen texts for both seen and unseen speakers.*

Input location	Gender-ind		Gender-dep	
	train	dev	train	dev
pre	0.357	0.402	0.438	0.361
attn	0.709	0.490	0.711	0.476
pre+attn	0.676	0.489	0.708	**0.533**
pre+attn+post	0.684	0.480	0.717	0.477

- Concatenate with encoder output only and input to attention mechanism (`attn`)
- Prenet + concatenate with encoder output (`pre+attn`)
- Prenet + concatenate with encoder output + postnet (`pre+attn+post`)

We did not try postnet input alone because we found that this configuration produced poor quality synthetic speech, but we decided to investigate its combination with other input locations.

Evaluation and Results: We objectively evaluated the different combinations of training strategy and embedding input locations by synthesizing some sample utterances from four "seen" speakers (those included in training) and four "unseen" ones (development speakers). Since we did not hold out any data from the "seen" speakers' utterances, we synthesized seen speakers' sample utterances from a randomly selected set of texts from the test set (unseen during training). We then extracted x-vectors for each speaker from the *synthesized* speech, and measured cosine similarity to the target speaker's x-vector extracted from his or her actual speech. Cosine similarity is defined as $cos\_sim(A, B) = A \cdot B / \|A\| \|B\|$ and is a standard measure of similarity of speaker embedding vectors for ASV. The values range from -1 to 1, and higher values indicate that the vectors are more similar. Cosine similarity results for different configurations are listed in Table 2.

As expected, we see a gap between seen and unseen speakers: seen speakers' synthetic speech generally has higher similarity to the original speech. Since the gender-dependent training with x-vectors input at both the prenet and attention mechanism produced the synthetic speech with best similarity for unseen speakers, we chose this configuration for our later experiments.

4.3. Comparing Different Embeddings for Speaker Similarity

After we chose the best training and model settings (gender-dependent training with embedding input at both the prenet and attention mechanism), we trained 15 TTS models each using a different type of speaker embedding: the 14 types of LDE embeddings described in Section 4.1 and x-vectors. We then conducted a crowdsourced listening test to evaluate naturalness and speaker similarity for both seen and unseen speakers using each speaker embedding as well as copy-synthesized speech and natural speech for comparison.

For each TTS system, we synthesized 50 sentences from each of the four "seen" (training) and eight "unseen" (development and test) speakers, for a total of 600 unseen test utterances per system. Listeners heard one test utterance at a time and first rated it on a Likert scale from 1-5 for Mean Opinion Score (MOS) for naturalness, then rated for speaker similarity compared to a reference utterance on a Differential MOS (DMOS) scale [41] from 1 (definitely a different speaker) to 4 (definitely the same speaker). Reference utterances were randomly chosen from the target speaker's original speech. Listeners rated "sets" of 25 utterances, and each listener could complete a maximum of ten sets. Each set was completed by five different listeners. Sets were designed to contain at least one utterance from every system to average out listener differences over all systems. Results from the listening test are in Table 3. Natural speech was rated with MOS of 3.83 and DMOS of 3.25.

Table 3. *MOS and DMOS results for seen (train) and unseen (dev and eval) speakers using each type of speaker embedding. Waveform generation was done using unseen texts for all speakers. Five-point and four-point scales were used for naturalness and similarity evaluation, respectively. Blue boxes show the best results for each condition and red boxes show second and third best.*

	Naturalness			Similarity		
system	train	dev	test	train	dev	test
vocoded	3.51	3.41	3.55	3.02	2.79	2.82
x-Vec$^{\mathbf{N}}$	3.20	3.19	3.19	2.93	1.86	2.37
LDE-1	3.15	3.16	3.21	2.87	**2.05**	2.34
LDE-1$^{\mathbf{N}}$	3.04	3.13	**3.46**	2.87	1.97	**2.45**
LDE-2	3.11	**3.28**	3.35	2.84	2.00	2.37
LDE-2$^{\mathbf{N}}$	3.13	3.19	3.33	2.90	2.00	2.35
LDE-3	3.09	3.24	**3.48**	2.89	1.88	**2.46**
LDE-3$^{\mathbf{N}}$	3.14	3.16	3.33	2.91	2.00	2.37
LDE-4	3.08	3.10	3.29	2.94	2.00	2.31
LDE-4$^{\mathbf{N}}$	3.12	3.20	3.29	2.90	1.98	2.39
LDE-5	3.07	**3.26**	**3.40**	2.89	1.99	**2.45**
LDE-5$^{\mathbf{N}}$	3.11	3.07	3.37	2.88	**2.02**	2.41
LDE-6	3.12	3.25	3.33	2.92	1.95	2.43
LDE-6$^{\mathbf{N}}$	3.13	**3.29**	3.23	2.88	1.94	2.39
LDE-7	3.15	3.03	3.18	2.91	1.86	2.28
LDE-7$^{\mathbf{N}}$	3.07	3.02	3.24	2.83	**2.02**	2.42

We found that speaker similarity scores for speakers seen during training are very close to those for vocoded speech. Similarity scores for unseen speakers (dev and test) are also lower than seen speakers, as expected, and consistent with Table 2. We observed that advanced neural speaker embeddings improve speaker similarity for unseen speakers compared to x-vectors. Unexpectedly, they also improve naturalness. While LDE helps, the impact of angular softmax and postprocessing (N) seems to be small. For completely unseen test set speakers, system LDE-3 was best in terms of both naturalness and speaker similarity. This system was significantly better than the x-vector system according to a Mann-Whitney U test both in terms of naturalness (p=5.9e-11) and speaker similarity (p=0.02). This was also the best type of embedding in terms of EER. We did not find any meaningful correlations between ASV and TTS scores.

5. CONCLUSIONS

We found that the LDE-based neural speaker embeddings can improve speaker similarity and naturalness of synthetic speech for unseen speakers, and this approach can be used for zero-shot speaker adaptation. However, there is still a gap between seen and unseen speaker similarity, indicating that the TTS model may still be overfitting to seen speakers and there is room for improvement. For future work, we will explore ways to mitigate this overfitting by trying different methods of speaker space augmentation. We would also like to evaluate adaptation performance on more nuanced aspects of speaker similarity, such as dialect and speaking style.

Acknowledgments This work was partially supported by a JST CREST Grant (JPMJCR18A6, VoicePersonae project), Japan, and by MEXT KAKENHI Grants (16H06302, 17H04687, 18H04120, 18H04112, 18KT0051, 19K24372), Japan. The numerical calculations were carried out on the TSUBAME 3.0 supercomputer at the Tokyo Institute of Technology.

6. REFERENCES

[1] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, Z. Y. Jaitly, Y. Xiao, Z. Chen, S. Bengio, Q. Le *et al.*, "Tacotron: Towards end-to-end speech synthesis," in *INTERSPEECH*, 2017.

[2] W. Ping, K. Peng, and J. Chen, "Clarinet: Parallel wave generation in end-to-end text-to-speech," *ICLR*, 2018.

[3] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerry-Ryan *et al.*, "Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions," *ICASSP*, 2018.

[4] J. Park, K. Zhao, K. Peng, and W. Ping, "Multi-speaker end-to-end speech synthesis," *CoRR*, vol. abs/1907.04462, 2019.

[5] Z. Kons, S. Shechtman, A. Sorin, C. Rabinovitz, and R. Hoory, "High quality, lightweight and adaptable TTS using LPCNet," *INTERSPEECH*, 2019.

[6] Y. Chen, Y. Assael, B. Shillingford, D. Budden, S. Reed, H. Zen, Q. Wang, L. C. Cobo, A. Trask, B. Laurie *et al.*, "Sample efficient adaptive text-to-speech," *ICLR*, 2019.

[7] B. Bollepalli, L. Juvela, and P. Alku, "Lombard speech synthesis using transfer learning in a Tacotron text-to-speech system," *INTERSPEECH*, 2019.

[8] J. Park, K. Zhao, K. Peng, and W. Ping, "Multi-speaker end-to-end speech synthesis," *arXiv preprint arXiv:1907.04462*.

[9] E. Nachmani, A. Polyak, Y. Taigman, and L. Wolf, "Fitting new speakers based on a short untranscribed sample," *ICML*, 2018.

[10] Y. Deng, L. He, and F. Soong, "Modeling multi-speaker latent space to improve neural TTS: Quick enrolling new speaker and enhancing premium voice," *arXiv preprint arXiv:1812.05253*, 2018.

[11] Q. Hu, E. Marchi, D. Winarsky, Y. Stylianou, D. Naik, and S. Kajarekar, "Neural text-to-speech adaptation from low quality public recordings," *Speech Synthesis Workshop 10*, 2019.

[12] S. Arık, J. Chen, K. Peng, W. Ping, and Y. Zhou, "Neural voice cloning with a few samples," in *Advances in Neural Information Processing Systems*, 2018, pp. 10 019–10 029.

[13] Y. Jia, Y. Zhang, R. Weiss, Q. Wang, J. Shen, F. Ren, P. Nguyen, R. Pang, I. L. Moreno, Y. Wu *et al.*, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," in *Advances in neural information processing systems*, 2018, pp. 4480–4490.

[14] S. Pascual, M. Ravanelli, J. Serrà, A. Bonafonte, and Y. Bengio, "Learning problem-agnostic speech representations from multiple self-supervised tasks," *INTERSPEECH*, 2019.

[15] M. Chen, M. Chen, S. Liang, J. Ma, L. Chen, S. Wang, and J. Xiao, "Cross-lingual, multi-speaker text-to-speech synthesis using neural speaker embedding," *INTERSPEECH*, 2019.

[16] J. Villalba, N. Chen, D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, J. Borgstrom, F. Richardson, S. Shon, F. Grondin *et al.*, "State-of-the-art speaker recognition for telephone and video speech: the JHU-MIT submission for NIST SRE18," *INTERSPEECH*, 2019.

[17] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," *ICASSP*, 2018.

[18] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2010.

[19] W. Cai, J. Chen, and M. Li, "Exploring the encoding layer and loss function in end-to-end speaker and language recognition system," *Odyssey 2018, The Speaker and Language Recognition Workshop*, 2018.

[20] Z. Huang, S. Wang, and K. Yu, "Angular softmax for short-duration text-independent speaker verification," in *Proc. Interspeech 2018*, 2018, pp. 3623–3627.

[21] N. Chen, J. Villalba, and N. Dehak, "Tied mixture of factor analyzers layer to combine frame level representations in neural speaker embeddings," *INTERSPEECH*, 2019.

[22] W. Xie, A. Nagrani, J. S. Chung, and A. Zisserman, "Utterance-level aggregation for speaker recognition in the wild," *ICASSP*, 2019.

[23] M. Hajibabaei and D. Dai, "Unified hypersphere embedding for speaker recognition," *arXiv preprint arXiv:1807.08312*, 2018.

[24] X. Xiang, S. Wang, H. Huang, Y. Qian, and K. Yu, "Margin matters: Towards more discriminative deep neural network embeddings for speaker recognition," *arXiv preprint arXiv:1906.07317*, 2019.

[25] Y. Jung, Y. Kim, H. Lim, Y. Choi, and H. Kim, "Spatial pyramid encoding with convex length normalization for text-independent speaker verification," *arXiv preprint arXiv:1906.08333*, 2019.

[26] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: a large-scale speaker identification dataset," *arXiv preprint arXiv:1706.08612*, 2017.

[27] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep speaker recognition," *arXiv preprint arXiv:1806.05622*, 2018.

[28] Y. Yasuda, X. Wang, S. Takaki, and J. Yamagishi, "Investigation of enhanced Tacotron text-to-speech synthesis systems with self-attention for pitch accent language," *ICASSP*, 2019.

[29] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital signal processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[30] G. Bhattacharya, J. Alam, and P. Kenny, "Deep speaker recognition: Modular or monolithic?" *Proc. Interspeech 2019*, pp. 1143–1147, 2019.

[31] A. W. Black and K. A. Lenzo, "Flite: a small fast run-time synthesis engine," in *4th ISCA Tutorial and Research Workshop (ITRW) on Speech Synthesis*, 2001.

[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. NIPS*, 2017, pp. 6000–6010. [Online]. Available: http://papers.nips.cc/paper/7181-attention-is-all-you-need

[33] J.-X. Zhang, Z.-H. Ling, and L.-R. Dai, "Forward attention in sequence-to-sequence acoustic modeling for speech synthesis," in *Proc. ICASSP*. IEEE, 2018, pp. 4789–4793.

[34] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," in *9th ISCA Speech Synthesis Workshop*, 2016.

[35] P. Kenny, T. Stafylakis, P. Ouellet, J. Alam, and P. Dumouchel, "PLDA for speaker verification with utterances of arbitrary duration," in *Proc. ICASSP*, 2013.

[36] N. Brümmer and J. du Preez, "Application-independent evaluation of speaker detection," *Computer Speech & Language*, vol. 20, no. 2, pp. 230 – 275, 2006.

[37] C. Veaux, J. Yamagishi, and K. MacDonald, "CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit," University of Edinburgh, The Centre for Speech Technology Research (CSTR), 2017.

[38] International Telecommunication Union, Recommendation G.191: Software Tools and Audio Coding Standardization, Nov 11 2005.

[39] S. King and V. Karaiskos, "The blizzard challenge 2011," in *Blizzard Challenge Workshop*, 2011.

[40] J. Taylor and K. Richmond, "Analysis of pronunciation learning in end-to-end speech synthesis," *INTERSPEECH*, 2019.

[41] J. Lorenzo-Trueba, J. Yamagishi, T. Toda, D. Saito, F. Villavicencio, T. Kinnunen, and Z. Ling, "The voice conversion challenge 2018: Promoting development of parallel and nonparallel methods," in *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018, pp. 195–202.