INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Semi-supervised Learning for Multi-speaker Text-to-speech Synthesis Using Discrete Speech Representation

Tao Tu, Yuan-Jui Chen, Alexander H. Liu, Hung-yi Lee

College of Electrical Engineering and Computer Science, National Taiwan University
{r07922022, r07922070, r07922013, hungyilee}@ntu.edu.tw

Abstract

Recently, end-to-end multi-speaker text-to-speech (TTS) systems gain success in the situation where a lot of high-quality speech plus their corresponding transcriptions are available. However, laborious paired data collection processes prevent many institutes from building multi-speaker TTS systems of great performance. In this work, we propose a semi-supervised learning approach for multi-speaker TTS. A multi-speaker TTS model can learn from the untranscribed audio via the proposed encoder-decoder framework with discrete speech representation. The experiment results demonstrate that with only an hour of paired speech data, whether the paired data is from multiple speakers or a single speaker, the proposed model can generate intelligible speech in different voices. We found the model can benefit from the proposed semi-supervised learning approach even when part of the unpaired speech data is noisy. In addition, our analysis reveals that different speaker characteristics of the paired data have an impact on the effectiveness of semi-supervised TTS.

Index Terms: multi-speaker speech synthesis, semi-supervised learning, discrete speech representation

1. Introduction

Recent advances in the neural-based end-to-end text-to-speech (TTS) systems have closed the gaps between the human speech and synthesized speech in the aspects of both speech quality and speech intelligibility [1, 2]. The notable results are shown not only for single speaker TTS modeling [3, 4, 5, 6, 7], multi-speaker TTS modeling [8, 9] but also for cloning prosody style [10, 11, 12, 13, 14] and speaker characteristics [15, 16, 17, 18]. However, these achievements require large amounts of paired high-quality speech and text data (i.e. paired data), which is typically unavailable under the low-resource condition due to the laborious and expensive data collection and human labeling. Contrarily, unannotated speech data (i.e. unpaired data) is relatively prevalent and accessible. Therefore, semi-supervised training of TTS that incorporates unpaired speech is crucial and worth investigating as it reduces the required amount of paired data for building a TTS system of high performance.

Semi-supervised learning for TTS has shown remarkable results in single speaker synthesis [19, 20, 21], where unpaired text or speech data are utilized to help the model training. Ren et al. [19] proposed to jointly train a phoneme recognition model and a speech synthesis model with unpaired data. Chung et al. [20] performed semi-supervised training on Tacotron [3] in a pretrain-finetune manner. Different from the pretrain-finetune method, Liu and Tu et al. [21] utilized the unpaired data for TTS training in an end-to-end manner. They proposed Sequential Representation Quantization AutoEncoder (SeqRQ-AE) to learn discrete speech representation from a large amount of unpaired speech data. With the aid from a few paired data, the discrete representations could be mapped to phonemes, and the model can be used for text-to-speech synthesis.

Even though many efforts have been made to semi-supervised learning for TTS, prior works [19, 20, 21] focused on single speaker TTS modeling and left multi-speaker TTS unstudied. Moreover, previous works [19, 21] leverage only a large amount of unpaired speech from a single speaker which is also challenging to collect in practice.

In this work, we move further to exploit semi-supervised multi-speaker TTS that can utilize unpaired speech with the supervision of only a few paired data (1 hour in total from either single speaker or multiple speakers). We propose an extended architecture of SeqRQ-AE for semi-supervised multi-speaker TTS, where our framework consists of a phonetic encoder as in SeqRQ-AE, an extended speaker representation table, and a multi-speaker TTS model as the decoder. The phonetic encoder transforms an utterance into a sequence of discrete phonetic representations by representation discretization and discrete representation mapping as in SeqRQ-AE. The speaker representation table contains speaker representation for each speaker in the training set. The decoder takes the phonetic representations along with the speaker representation and decodes the corresponding speech. This encoder-decoder framework can be jointly learned from unpaired data by imposing a reconstruction loss. Samples drawn from our model are provided on `https://ttaoretw.github.io/multispkr-semi-tts/demo.html`.

The contributions of this paper are highlighted as follows:

- To the best of our knowledge, this is the first study of semi-supervised multi-speaker TTS.
- Our semi-supervised method matches the performance of the fully-supervised topline when only 1 hour of multi-speaker training data is annotated.
- When only 1 hour of single-speaker training data is labeled, our method can still generate intelligible speech of different voices.
- The effectiveness of semi-supervised multi-speaker TTS is further verified by considering the experiment with noisy unpaired speech data, which makes our method more feasible in practice.
- We take a closer look at the impact of speaker characteristics on the effectiveness of semi-supervised TTS.

2. Sequential Representation Quantization AutoEncoder (SeqRQ-AE)

In this section, we briefly overview the SeqRQ-AE, which is trained from a large amount of unpaired audio X_{unpair} and limited audio-text pairs $(X_{\text{pair}}, Y_{\text{pair}})$, where Y_{pair} is the corresponding phoneme sequence of X_{pair}.

http://dx.doi.org/10.21437/Interspeech.2020-1824

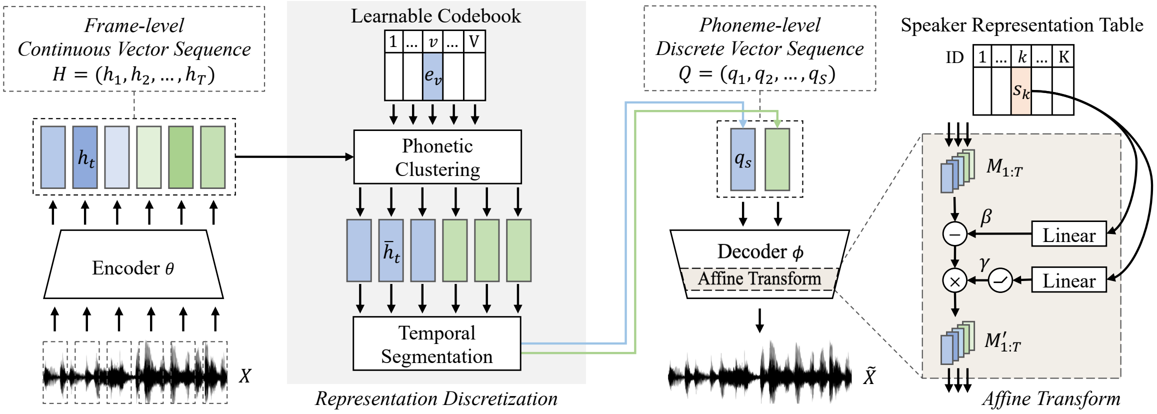

Figure 1: *Overview of the proposed model. The input speech X is first encoded into the frame-level continuous vector sequence H. Next, Representation Discretization (see Sec. 2.1) is performed to obtain the phoneme-level discrete vector sequence Q. Q would be fed into a sequence-to-sequence decoder conditioned on speaker representation to reconstruct the input speech (see Sec. 3).*

2.1. Phonetic Encoder

Given a speech sequence $X = (x_1, x_2, \ldots, x_T)$ of T frames, an encoder network $\text{Enc}_\theta(\cdot)$ extracts the corresponding frame-level representation sequence

$$H \equiv (h_1, h_2, \ldots, h_T) = \text{Enc}_\theta(X). \quad (1)$$

To obtain the phoneme-level discrete speech representation sequence $Q = (q_1, q_2, \ldots, q_S)$ that matches the underlying phoneme sequence, *representation discretization* and *discrete representation mapping* are applied.

Representation Discretization. To perform representation discretization, a learnable *codebook* $E = (e_1, e_2, \ldots, e_V)$ of size V is maintained, where each $e_i \in \mathbb{R}^D$ is called a *codeword*. For an encoded frame-level representation sequence H, the closest codeword e_v is used as a substitute for each representation h_t, and this operation is called *phonetic clustering* [21]. The gradient of this non-differentiable operation is approximated by straight-through (ST) gradient estimator [22]. The phonetic clustering process produces a codeword sequence $\bar{H} = (\bar{h}_1, \bar{h}_2, \ldots, \bar{h}_T)$ of length T where each element is one of the V codewords. Besides, to match the sequence length of the codeword sequence to the underlying phoneme sequence, *temporal segmentation* is performed to group repeated consecutive codewords into one codeword.

Discrete Representation Mapping. To force each code of the codebook to be a phoneme, we first set the codebook size V to be the number of all phonemes and assign each entry e_v a phoneme v. The paired speech data (X_{pair}, Y_{pair}) is used for learning the mapping. The probability of a continuous representation h_t being mapped to a codeword e_v is defined as

$$P(v|h_t) = \frac{\exp(-\|h_t - e_v\|_2)}{\sum_{k \in V} \exp(-\|h_t - e_k\|_2)}, \quad (2)$$

and the probability for a frame-level phoneme sequence $\tilde{Y} = (v_1, v_2, \ldots, v_T)$ can be approximated by

$$P(\tilde{Y}|H) \approx \prod_{t=1}^{T} P(v_t|h_t). \quad (3)$$

Then, the connectionist temporal classification [24] (CTC) is applied on the paired data with Eq. (3) to maximize the log-likelihood of outputting target Y_{pair}.

2.2. Speech Synthesizer

To reconstruct the input utterance, a decoder network $\text{Dec}_\phi(\cdot)$ takes the sequence of discrete speech representations Q as input and synthesize audio $\widetilde{X}$ as below.

$$\widetilde{X} = \text{Dec}_\phi(Q). \quad (4)$$

In addition, the decoder can also do text-to-speech transformation by inputting code sequence Q_{pair} retrieved from the codebook according to the ground truth phoneme sequence Y_{pair}. The overall loss function can be written as

$$\begin{aligned} L_{\text{total}} = &\ \lambda \cdot \text{MSE}(\widetilde{X}, X_{\text{unpair}}) \\ &- \log P(Y_{\text{pair}}|H) \\ &+ \text{MSE}(\text{Dec}_\phi(Q_{\text{pair}}), X_{\text{pair}}), \end{aligned} \quad (5)$$

where the first term is the reconstruction loss of unpaired speech X_{unpair}, the second term is the CTC loss for Y_{pair}, the last term is the TTS loss for target audio X_{pair}, and λ is fixed to be 10 throughout the end-to-end training process. For more details, please refer to the prior work [21].

3. Multi-speaker SeqRQ-AE

In the previous work [21], both X_{unpair} and X_{pair} are produced by the same speaker. Here we assume the audio is from multiple speakers[1], and we extend the decoder in Sec. 2.2 into a multi-speaker synthesizer.

In order to perform multi-speaker synthesis (as shown in the right-hand side of Figure 1), the decoding process in Eq. (4) is equipped with a learnable speaker representation table $\{s_1, \ldots, s_k, \ldots, s_K\}$, where each vector s_k is the embedding of a speaker, and K is the total number of speakers in X_{pair} and X_{unpair}. With speaker representations, Eq. (4) is modified as below:

$$\widetilde{X} = \text{Dec}_\phi(Q, s_k), \quad (6)$$

where the decoder is conditioned on the speaker representation s_k obtained from the speaker representation table according to speaker identity of the input utterance. The loss function to be optimized is the same as Eq. (5), except that $\text{Dec}_\phi(Q)$ is replaced with $\text{Dec}_\phi(Q, s_k)$.

[1] We assume the speaker identities of both the paired and unpaired audio data are known.

Table 1: *Performance comparison of different methods. The subscript "(n)" indicates the MUSAN noises are added to the speech data. The naturalness MOS (i) is reported with 95% confidence intervals. The recognition result (ii) is reported with character error rate (CER).*

Experiment	**Method**	**Paired Data**	**Multi-speaker Supervised**	**Unpaired Data**	**(i) Naturalness**	**(ii) CER**
Baseline	(b-1) Ground Truth	-	-	-	4.88 ± 0.033	7.98
	(b-2) Tacotron-2	VCTK-25_{hr}	✓(108_{spkr})	-	3.59 ± 0.066	8.11
	(b-3) Tacotron-2	VCTK-1_{hr}	✓(108_{spkr})	-	1.47 ± 0.055	72.67
Semi-supervised	(s-1) Sp-chain [23]†	VCTK-1_{hr}	✓(50_{spkr})	VCTK-25_{hr}-108_{spkr} + LJ-other	2.81 ± 0.071	31.30
	(s-2) Ours	VCTK-1_{hr}	✓(50_{spkr})	VCTK-25_{hr}-108_{spkr} + LJ-other	3.46 ± 0.066	11.53
	(s-3) Sp-chain	LJ-1_{hr}	-	VCTK-25_{hr}-108_{spkr} + LJ-other	2.10 ± 0.065	45.47
	(s-4) Ours	LJ-1_{hr}	-	VCTK-25_{hr}-108_{spkr} + LJ-other	3.09 ± 0.073	21.70
- w/ Noise	(n-1) Ours	VCTK-1_{hr}	✓(50_{spkr})	VCTK-14_{hr}-60_{spkr} + LJ-other	2.02 ± 0.087	41.95
	(n-2) Ours	VCTK-1_{hr}	✓(50_{spkr})	VCTK-14_{hr}-60_{spkr} + VCTK$_{(n)}$-11_{hr}-48_{spkr} + LJ-other	3.28 ± 0.073	12.78
	(n-3) Ours	LJ-1_{hr}	-	VCTK-14_{hr}-60_{spkr} + LJ-other	1.61 ± 0.069	80.07
	(n-4) Ours	LJ-1_{hr}	-	VCTK-14_{hr}-60_{spkr} + VCTK$_{(n)}$-11_{hr}-48_{spkr} + LJ-other	2.85 ± 0.070	21.85
- w/ different Characteristics	(c-1) Ours	LJ-1_{hr}	-	VCTK-25_{hr}-108_{spkr} + LJ-other + MLJ-other	3.22 ± 0.070	16.47
	(c-2) Ours	MLJ-1_{hr}	-	VCTK-25_{hr}-108_{spkr} + LJ-other + MLJ-other	2.31 ± 0.062	15.36
	(c-3) Ours	FLJ-1_{hr}	-	VCTK-25_{hr}-108_{spkr} + FLJ-other + MLJ-other	3.07 ± 0.079	16.20

† Trained without text-to-text cycle.

To perform speaker adaptive synthesis, we proposed to modify the intermediate state of the decoder with an affine transformation. The scaling factor γ and the shifting magnitude β for some particular speaker k can be derived by

$$\begin{aligned} \gamma &= \text{ReLU}(W_\gamma s_k + b_\gamma), \\ \beta &= W_\beta s_k + b_\beta, \end{aligned} \tag{7}$$

where s_k is the speaker representation of speaker k and both W, b are learnable parameters of linear projection layer. With the scaling factor γ and the shifting magnitude β, the affine transformation is performed on the intermediate state M_t of the decoder at each timestep t of the synthesis process

$$M_t^{'} = \gamma(M_t - \beta). \tag{8}$$

In practice, we employ Tacotron-2 [4] as the decoder $\text{Dec}_\phi(\cdot)$ of our framework, where Tacotron-2 itself contains an encoder (Taco-encoder) and a decoder (Taco-decoder). Taco-decoder consists of 2 LSTM layers and 5 convolution layers, where we select the output hidden states of the first LSTM as the input of affine transformation M_t. Afterward, the modified LSTM output $M_t^{'}$ is passed to the next layer. We found that this affine transformation scheme makes the training of the multi-speaker TTS model more stable.

4. Experiment

4.1. Experiment Setup

Model Architecture. For the phonetic encoder and the codebook, we follow the setup as in the prior work [21]. The Griffin-Lim algorithm [25] is applied to estimate the phases and converts spectrograms to waveforms as in Tacotron [3]. The differential spectral loss [26] is also adopted to boost the performance of the TTS model.

Datasets We use VCTK corpus [27], which consists of read English speech data from 108 speakers with complete transcriptions. After removing the leading and ending silence by Montreal Forced Aligner [28], we have about 26 hours of speech data in total. We randomly choose 1000 audio files for testing and other 1000 audio for selecting hyperparameters. In addition, an hour data randomly chosen from LJSpeech [29], which is a 24 hours English dataset from a single female speaker, is used as the paired speech data (LJ-1_{hr}) and the remaining data are used as unpaired speech data (LJ-other). Moreover, we use Google cloud text-to-speech to synthesize MLJ and FLJ datasets based on the text from LJSpeech [29], where MLJ and FLJ are from a male and a female speaker, respectively. Following the data partition of LJSpeech, MLJ and FLJ are also split into MLJ-1_{hr}, FLJ-1_{hr} and MLJ-other, FLJ-other. We use x_{hr} to indicate the total amount of audio data (in hours), and x_{spkr} to indicate the number of used speakers where speech data size for each speaker is roughly equivalent. The speakers in the test set will not appear in the paired training data set for all experiments except for the baseline experiments (b-2) (b-3). As for text and audio preprocessing, we follow the prior work [21].

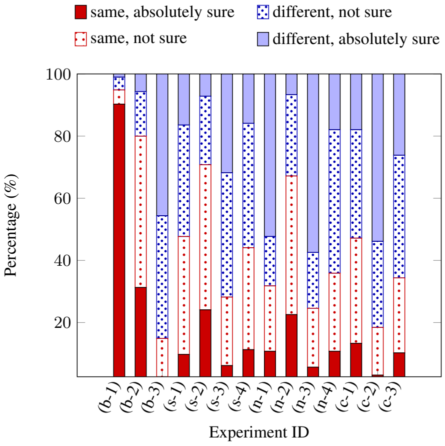

Figure 2: *The results of speaker similarity test. The x-axis labels indicate the experiment IDs as in Table 1.*

Speech Naturalness Test. The Mean Opinion Score (MOS) test is conducted for measuring speech quality, where 50 sentences are randomly chosen from the testing set and listened by 60 subjects. The subjects are asked to rate the audio based on the speech naturalness. The rating is on a 5-point scale in increments of 1. The higher the MOS, the better the quality of the given audio. Each audio file receives at least 6 ratings. The results are shown in the col. (i) of Table 1.

Content Correctness Test. To analyze whether the model outputs correct speech content, we conduct Automatic Speech

Recognition (ASR) test using the ASR service provided by Google Cloud Speech API to recognize synthesized audio or ground truth audio in the testing set. Then, the character error rate (CER) is computed based on the ground truth texts. The lower the CER, the more accurate the model output content is. The results are shown in the col. (ii) of Table 1.

Speaker Similarity Test. To measure the speaker similarity, the speaker similarity test [30] is conducted. Given a pair of ground truth audio sample and TTS output sample from the same speaker with different contents, 60 subjects were asked to answer the question: "Do you think the same speaker has produced these two samples?" with options "same, absolutely sure", "same, not sure", "different, not sure" and "different, absolutely sure". There are 50 randomly chosen pairs are answered by subjects, and each pair receives at least 6 answers. The results are shown in Figure 2.

Baseline. To objectively evaluate the effectiveness of our proposed method, we first perform the evaluation on raw audio from the test set and the original Tacotron-2 model with *full supervision* to serve as our baseline as shown in Table 1. The Tacotron-2 model that is fully supervised by 25 hours (b-2) and 1 hour (b-3) of multi-speaker paired data can be viewed as the top-line and the bottom-line performance of our semi-supervised methods, respectively.

4.2. Multi-speaker Speech Synthesis

In this part ("Semi-supervised" partition of Table 1), we also compared our method to the speech chain[2] [23] model (Sp-chain, row (s-1) and (s-3) in Table 1) that shares the same architecture with our proposed model. Sp-chain can be viewed as our model *without* the learnable codebook and ST gradient estimator for unpaired speech data.

Semi-supervised TTS w/ Multi-speaker Paired Data. In this setting, 1 hour of paired data comes from 50 speakers (about 72 seconds for each speaker) are utilized for the TTS training. For speech naturalness (Table 1 col. (i)), content correctness (Table 1 col. (ii)) , and speaker similarity (Figure 2), we can see that the speech quality of our method (s-2) is consistently better than Sp-chain (s-1). We conjecture this is because the proposed method allows the gradients to flow from the decoder through the encoder using the ST gradient estimator while Sp-chain does not. This makes the encoder and the codebook also being updated to obtain superior representations for better speech reconstruction. It is worth noticing that our method (s-2) *matches the performance of the topline (b-2)* which has seen the speakers in the test set. In the meanwhile, the bottom-line model (b-3) can hardly generate intelligible speech. This demonstrates the effectiveness of semi-supervised TTS training.

Semi-supervised TTS w/ Single Speaker Paired Data. In this setting, all paired data comes from a single speaker, which indicates that the model can only learn to synthesize different speaker from unpaired data. Consistent with the setting of multi-speaker paired data, our method (s-4) outperforms the Sp-chain (s-3) in this setting. By comparing (s-4) to (s-2), we find that the quality of synthesized speech deteriorates a bit when the paired data come from only one speaker. Despite the performance drop compared to multi-speaker paired data setting, the multi-speaker TTS model trained in single speaker paired data setting is still much better than the baseline (b-3) in both naturalness and speaker similarity, which demonstrates the gain from semi-supervised learning.

4.3. Impact of Noisy Unpaired Data

In this experiment, we discuss whether the proposed method can benefit from noisy unpaired data. This is important because considerable high-quality clean audio is hard to collect and unpaired data are likely to be recorded in noisy environments in the real world case. To simulate this situation, we take a part of VCTK data and manually add noise to it, which we refer to as VCTK$_{(n)}$ in the "w/ Noise" experiment of Table 1. The noises (10-30dB SNR) used in VCTK$_{(n)}$ are randomly selected from the MUSAN dataset [31]. This synthetic noisy dataset includes 11 hours of speech from 48 speakers while the rest 14 hours of speech from 60 speakers in VCTK remains clean without noise.

Results with noisy unpaired data are reported in the "w/ Noise" experiment of Table 1. For the model (n-1) trained with 14 hours of clean unpaired data, the quality of the output speech is worse than the output of the model (n-2) trained with additional noisy unpaired data. This can be seen from the speech naturalness, content correctness, and speaker similarity. Besides, by comparing (n-2) with (s-2), we can see that the synthesis performance only drops a bit when some part of the unpaired data is noisy. These demonstrate that the TTS model can benefit from unpaired data even if part of it is noisy. The experiments in single speaker paired data setting are also conducted here (n-3) (n-4) and the same conclusion can be obtained.

4.4. Impact of Speaker Characteristics of Paired Data

According to the result of the experiment (s-4), it is possible to construct a TTS model with only 1 hour paired data from a single speaker. In this section, we would like to further study *how the characteristics of the paired data influence the performance*. The results are reported in the "w/ different Characteristics" experiment of Table 1.

First, we conduct (c-1) experiment with the paired data LJ-1$_{\text{hr}}$ from a female speaker and (c-2) experiment with the paired data MLJ-1$_{\text{hr}}$ from a synthesized male speaker[3]. We can see that the performance of (c-2) drops a lot no both in naturalness or in speaker similarity.To verify this drop comes from gender or synthesized nature, we additionally change the data from LJ to FLJ which is from a synthesized female speaker (experiment (c-3)). We observe that (c-3) deteriorates slightly than (c-1) with respect to naturalness and speaker similarity, which implies that training with synthesized speech only slightly hurt and the performance decline of (c-2) mainly comes from its male characteristic. Therefore, we conclude that speaker characteristics are essential and female voice might be more applicable for semi-supervised learning when only single speaker paired data is available.

5. Conclusion

In this work, we study the semi-supervised multi-speaker TTS. Experiments show our proposed semi-supervised method matches the performance of the fully-supervised topline. In the future, we aim to explore the usage of the proposed method in cross-lingual settings [32, 33].

[2] Text-to-text cycle is not used since we found that the text-to-text cycle hurts the performance a lot when there is only one hour paired data available.

[3] Synthesized audio is used here because we do not have large enough labeled clean audio from one male speaker.

6. References

[1] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[2] W. Ping, K. Peng, and J. Chen, "Clarinet: Parallel wave generation in end-to-end text-to-speech," *arXiv preprint arXiv:1807.07281*, 2018.

[3] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio *et al.*, "Tacotron: Towards end-to-end speech synthesis," *Proc. Interspeech 2017*, pp. 4006–4010, 2017.

[4] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.

[5] Y. Taigman, L. Wolf, A. Polyak, and E. Nachmani, "Voiceloop: Voice fitting and synthesis via a phonological loop," *arXiv preprint arXiv:1707.06588*, 2017.

[6] J. Sotelo, S. Mehri, K. Kumar, J. F. Santos, K. Kastner, A. Courville, and Y. Bengio, "Char2wav: End-to-end speech synthesis," 2017.

[7] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech: Fast, robust and controllable text to speech," in *Advances in Neural Information Processing Systems*, 2019, pp. 3165–3174.

[8] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep voice 3: Scaling text-to-speech with convolutional sequence learning," *arXiv preprint arXiv:1710.07654*, 2017.

[9] J. Park, K. Zhao, K. Peng, and W. Ping, "Multi-speaker end-to-end speech synthesis," *arXiv preprint arXiv:1907.04462*, 2019.

[10] V. Wan, C.-a. Chan, T. Kenter, J. Vit, and R. Clark, "Chive: Varying prosody in speech synthesis with a linguistically driven dynamic hierarchical conditional variational network," *arXiv preprint arXiv:1905.07195*, 2019.

[11] A. Rosenberg, B. Ramabhadran, G. Sun, H. Zen, R. J. Weiss, Y. Wu, Y. Zhang, and Y. Cao, "Generating diverse and natural text-to-speech samples using quantized fine-grained vae and autoregressive prosody prior," in *ICASSP*, 2020.

[12] V. Klimkov, S. Ronanki, J. Rohnke, and T. Drugman, "Fine-grained robust prosody transfer for single-speaker neural text-to-speech," *arXiv preprint arXiv:1907.02479*, 2019.

[13] Y. Lee and T. Kim, "Robust and fine-grained prosody control of end-to-end speech synthesis," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5911–5915.

[14] R. Skerry-Ryan, E. Battenberg, Y. Xiao, Y. Wang, D. Stanton, J. Shor, R. J. Weiss, R. Clark, and R. A. Saurous, "Towards end-to-end prosody transfer for expressive speech synthesis with tacotron," *arXiv preprint arXiv:1803.09047*, 2018.

[15] Y. Wang, D. Stanton, Y. Zhang, R. Skerry-Ryan, E. Battenberg, J. Shor, Y. Xiao, F. Ren, Y. Jia, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," *arXiv preprint arXiv:1803.09017*, 2018.

[16] W.-N. Hsu, Y. Zhang, R. J. Weiss, H. Zen, Y. Wu, Y. Wang, Y. Cao, Y. Jia, Z. Chen, J. Shen *et al.*, "Hierarchical generative modeling for controllable speech synthesis," *arXiv preprint arXiv:1810.07217*, 2018.

[17] 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.

[18] Y. Zhang, R. J. Weiss, H. Zen, Y. Wu, Z. Chen, R. Skerry-Ryan, Y. Jia, A. Rosenberg, and B. Ramabhadran, "Learning to speak fluently in a foreign language: Multilingual speech synthesis and cross-language voice cloning," *arXiv preprint arXiv:1907.04448*, 2019.

[19] Y. Ren, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Almost unsupervised text to speech and automatic speech recognition," *arXiv preprint arXiv:1905.06791*, 2019.

[20] Y.-A. Chung, Y. Wang, W.-N. Hsu, Y. Zhang, and R. Skerry-Ryan, "Semi-supervised training for improving data efficiency in end-to-end speech synthesis," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6940–6944.

[21] A. H. Liu, T. Tu, H.-y. Lee, and L.-s. Lee, "Towards unsupervised speech recognition and synthesis with quantized speech representation learning," *arXiv preprint arXiv:1910.12729*, 2019.

[22] Y. Bengio, N. Léonard, and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," *arXiv preprint arXiv:1308.3432*, 2013.

[23] A. Tjandra, S. Sakti, and S. Nakamura, "Listening while speaking: Speech chain by deep learning," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 301–308.

[24] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 369–376.

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

[26] S. Shechtman and A. Sorin, "Sequence to sequence neural speech synthesis with prosody modification capabilities," *arXiv preprint arXiv:1909.10302*, 2019.

[27] 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.

[28] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal forced aligner: Trainable text-speech alignment using kaldi." 2017.

[29] K. Ito, "The lj speech dataset," https://keithito.com/LJ-Speech-Dataset/, 2017.

[30] M. Wester, Z. Wu, and J. Yamagishi, "Analysis of the voice conversion challenge 2016 evaluation results." 2016.

[31] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[32] Y. Zhou, X. Tian, H. Xu, R. K. Das, and H. Li, "Cross-lingual voice conversion with bilingual phonetic posteriorgram and average modeling," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6790–6794.

[33] Y.-J. Chen, T. Tu, C.-c. Yeh, and H.-Y. Lee, "End-to-end text-to-speech for low-resource languages by cross-lingual transfer learning," *Proc. Interspeech 2019*, pp. 2075–2079, 2019.