PHONE-INFORMED REFINEMENT OF SYNTHESIZED MEL SPECTROGRAM FOR DATA AUGMENTATION IN SPEECH RECOGNITION

Sei Ueno, Tatsuya Kawahara

Graduate School of Informatics, Kyoto University, Sakyo-ku, Kyoto, Japan

ueno@sap.ist.i.kyoto-u.ac.jp

ABSTRACT

While recent end-to-end automatic speech recognition (ASR) models achieve high performance, we need to prepare an abundant amount of training data, which is a barrier to apply them to a specific domain. To mitigate the lack of training data, text-to-speech (TTS) systems have been utilized to leverage text-only data to efficiently generate paired data for training the ASR model. The widely-used procedure first generates a Mel spectrogram from text data, then converts it into a waveform, and converts it again to a Mel spectrogram. The vocoder is often used to alleviate the difference between real and synthesized speech, but it requires a huge amount of runtime. In this work, we propose a phone-informed post-processing network that refines Mel spectrograms without using the vocoder. The proposed network consumes not only Mel spectrograms but also text information to use phone sequence information for refinement. Experimental evaluations demonstrate that the proposed network achieves better WERs than the vocoder network in an English domain adaptation task (LibriSpeech to TED-LIUM 2; read speech to spontaneous speech) in a much smaller amount of data generation time. It is also shown the use of phone information is critical for the improvement. We also confirm the effect of the proposed model in a Japanese domain adaptation task (CSJ-SPS to CSJ-APS; everyday topic to academic topic).

Index Terms— Speech recognition, domain adaptation, speech synthesis, FastSpeech 2, Transformer

1. INTRODUCTION

The automatic speech recognition (ASR) models have recently achieved high performance because of the progress of deep learning. In particular, end-to-end models that integrate the acoustic and language models have realized better performance than the conventional hybrid systems. Among the end-to-end ASR models, the connectionist temporal classification (CTC)-based model [1], the attention-based encoder-decoder model [2], recurrent neural network transducer (RNN-T) [3], and transformer-based models [4, 5] have been investigated. However, we need a huge amount of paired speech and transcription data for training these models. It is not easy to prepare the paired-data for adapting to a specific domain.

On the other hand, it is often the case there are a huge amount of text-only data for the target domain. To compensate training data, many works leveraging text-only data have been pursued. In this work, we focus on an approach which generates paired data by a text-to-speech (TTS) system and then train an ASR model using real and generated data for data augmentations [6, 7, 8, 9, 10, 11, 12] or domain adaptation [13, 14, 15, 16]. This approach uses a text-to-mel system such as Tacotron 2 [17] to generate a Mel spectrogram from a text sequence. After generating Mel spectrograms, there are two major ways of data generation for ASR. One is to directly use the synthesized Mel spectrograms without any post-processing [8, 9, 10, 13, 14]. The other uses a vocoder to convert the Mel spectrogram into a waveform [6, 7, 11, 12, 15, 16], which is again converted into a Mel spectrogram used for the ASR input. The neural network-based vocoder is generally used since it delivers better performance than the conventional vocoders such as the Griffin-Lim algorithm [11]. The benefit of using the vocoder is that we can design ASR and TTS models independently since we can use an optimal setting for Mel spectrograms for respective systems. Moreover, the vocoder improves the quality of data and performance of augmentation compared with direct use of Mel spectrograms. In this way, the vocoder is regarded as a post-processing network for enhancing Mel spectrogram.

However, converting to a waveform needs an extra time for data augmentation. Moreover, synthesizing waveforms takes a huge amount of time because the ASR system needs a huge amount of training data, although it does not need waveforms. In this paper, we propose a phone-informed post-processing network to refine the synthesized Mel spectrograms. The proposed network focuses on filling the gap between real and synthesized Mel spectrograms and is used instead of the vocoder. Refinement on the Mel spectrogram takes less inference time than the vocoder synthesizing waveforms. For improved enhancement, we use the text information, specifically phone information of the speech, which is readily available in the TTS task.

In the rest of the paper, we first review related work in Section 2. Section 3 explains the proposed post-processing network. Experimental evaluations using two datasets are presented in Section 4.

2. RELATED WORK

2.1. Text-to-mel network and vocoder

We briefly review the neural network-based TTS model. It generally has two separated networks: a text-to-mel network and a vocoder (mel-to-waveform) network. The text-to-mel network generates a Mel spectrogram from a phone (or character) sequence. For this network, several models such as Tacotron 2 [17] and the Transformer-based model [18] have been investigated. These models generate a high-quality speech with a much simpler architecture compared with the conventional statistical models. Most recently, non-autoregressive networks have been proposed such as FastSpeech-1,2 [19, 20] and Parallel Tacotron-1,2 [21, 22]. In this work, we use the FastSpeech 2-based model because of the fast generation. For a non-autoregressive generation, the FastSpeech 2 model first predicts the duration of the Mel spectrogram corresponding to each phone in the variance adaptor and extends the phone embedding information for the duration of the segment. The model

then predicts the additional acoustic information such as F0 and energy, which are added to the extended phone information. These features are fed into the decoder to generate a Mel spectrogram.

On the other hand, the vocoder (mel-to-waveform) network converts the generated Mel spectrogram into a waveform. The models such as LPCNet [23] and MelGAN [24] are attractive for fast inference. However, the inference still needs much time since the waveform has many samples and often becomes a very long sequence.

2.2. Data generation for ASR using TTS

Many works have investigated the efficient use of generated speech for data augmentation and domain adaptation of ASR systems. Mimura *et al.* [13] fixed the ASR acoustic encoder in training with the synthesized data. Wang *et al.* [9, 10] investigated consistency regularization for TTS incorporated with ASR. Zheng *et al.* [15] introduced a loss for regularization of the decoder when the ASR model was finetuned for out-of-vocabulary words. Fazel *et al.* [12] investigated a multi-stage training strategy by combining weighted multi-style training, data augmentation, encoder freezing, and parameter regularization. Chen *et al.* [8] introduced a generative adversarial network (GAN)-based model for the pre-trained TTS and ASR to increase the acoustic diversity in the synthesized data. Kurata *et al.* [16] introduced a mapping network before the ASR encoder to convert the acoustic features of the synthesized audio to those of the target domain, which is similar to this study but did not use the phone information.

3. PROPOSED METHOD

3.1. Baseline architecture of data generation for ASR

For data augmentation for ASR, we compose a multi-speaker text-to-mel network, which is generally used [6, 7, 12, 14]. There are some options for the multi-speaker embeddings such as speaker IDs [14], variational autoencoder (VAE) latent variables [6], pre-trained speaker verification model [12], and a global style token (GST) [7]. In this work, we use a speaker ID embedding.

3.2. Phone-informed post-processing network for ASR

In the standard TTS task, the role of the vocoder is to generate a waveform that people can hear and evaluate. On the other hand, in the data augmentation task, the vocoder aims to fill the discrepancy between the Mel spectrogram settings of ASR and TTS without changing the input of each model. Moreover, the vocoder can alleviate the quality gap between the real and synthesized Mel spectrograms. We observe that synthesized Mel spectrogram become clear after applying the vocoder. However, the vocoder model takes a long time for inference. Moreover, the text-to-mel and vocoder models must be applied step by step. We also need to convert the waveform to a Mel spectrogram again.

In this work, we propose a phone-informed post-processing network instead of the vocoder, whose data generation time is much smaller than the vocoder. Specifically, we compose a mel-to-mel network to directly refine the synthesized Mel spectrogram and fill the gap from the real speech. Refining the speech on the Mel spectrograms domain takes less time than that on the waveform domain. For general speech enhancement, masking is widely applied to noisy (not Mel) spectrograms [25], but it cannot use text information because it is not usually available. However, it is well known that enhancement will be improved given phone information of the speech [26, 27], which is available in TTS and data augmentation tasks. Thus, we use phone embedding information.

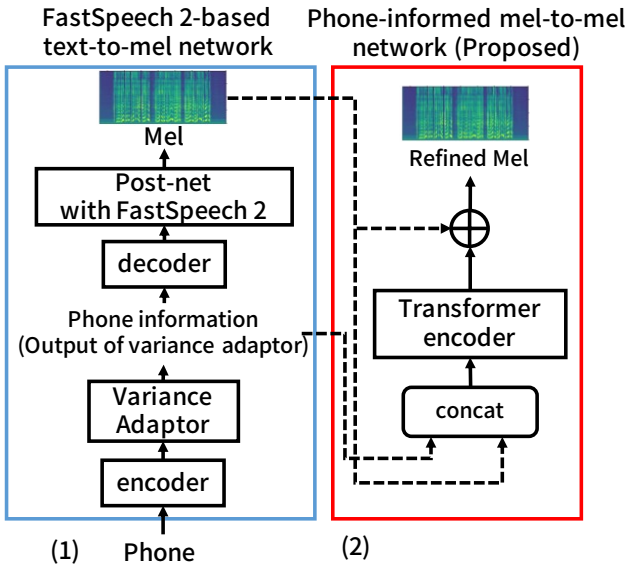

Fig. 1: The architecture of the proposed phone-informed post-processing network. (1) FastSpeech 2-based model. (2) Proposed post-processing network using the synthesized Mel spectrogram and phone information (the output of the variance adaptor).

Fig. 1 shows an architecture of the proposed phone-informed post-processing network. We train the FastSpeech 2-based model at the first stage. After training it, we do not update its parameters. Next, we compose a Transformer-based network that consumes the generated Mel spectrogram and the output of the variance adaptor which corresponds to phone embedding information. The generated Mel spectrogram and the output of the variance adaptor are taken from the FastSpeech 2-based model. The residual block is adopted in the proposed method as in the post-net [17] in FastSpeech 2. We use an L1 loss between the predicted and the ground-truth Mel spectrogram for the objective of training the proposed network. Although the FastSpeech 2-based model is trained on the same criteria, it must learn a complex mapping from a text to the Mel spectrogram together with the duration, pitch, and energy. On the other hand, the proposed post-processing network is expected to minimize the L1 loss more efficiently since it is given an approximate spectrogram. In training, we add the weighted loss separately for low (0-20) and high (21-80) frequencies. In this work, we used 1.4 and 0.6, respectively. Moreover, we also feed phone information, which is readily available, unlike general speech enhancement. However, the length of phone sequence is much shorter than that of the Mel spectrogram. In this work, we use the output of the variance adaptor in the FastSpeech 2 model, which predicts the duration of each phone and extends the outputs of the encoder to the duration. The output length of the variance adaptor is the same as the predicted Mel spectrogram length.

When we use the vocoder network, we can change the setting of the synthesized Mel spectrogram to that used in the ASR via a waveform. On the other hand, the proposed method needs to use

the same setting such as the FFT size, the frame length and shift[1]. However, recent ASR networks such as Transformer use some CNN sub-sampling layers, and thus the difference of the settings in TTS and ASR can be filled.

4. EXPERIMENT EVALUATIONS

4.1. Datasets and tasks

We conducted two domain adaptation experiments, one in English and the other in Japanese. In training the TTS and ASR models in English, we used LibriTTS [28] and LibriSpeech corpus [29]. LibriTTS is a sub-corpus of LibriSpeech designed for the TTS task. We downsampled waveforms of LibriTTS to match the sampling rate to 16kHz in all datasets. In LibriTTS and LibriSpeech, we used the train-clean-100 subset. The train-clean-100 of LibriSpeech contains 100 hours of speech data. The train-clean-100 of LibriTTS contains 53.8 hours of speech data including 247 speakers (Female: 123, Male: 124)

For the TTS model, a word sequence was converted into 85-class phones by an open-source grapheme-to-phone tool[2]. To obtain the alignment for training the variance adaptor, we also trained a CTC-based ASR model with the train-clean-100 and conducted forced alignment. A pitch (F0) was predicted by WORLD [30].

For ASR tasks, a word sequence was converted into 10k-class byte-pair-encoding (BPE) units. In this experiment, we suppose that speech generation is used for domain adaptation from read speech (LibriSpeech) to spontaneous speech. For the target domain, we used TED-LIUM release-2 corpus [31] of 91,967 utterances (211 hours). We used only transcription for generating speech. The generated speech was mixed with the real speech of LibriSpeech when training the ASR model. For language model integration, we used official TED-LIUM 2 text data.

In the task in Japanese, we used the CSJ [32], which has two different domain subsets named SPS (Simulated Public Speaking) and APS (Academic Presentation Speech). While SPS is speech on everyday topics, APS is live recordings of academic presentations. SPS has 324.1 hours of speech including 1704 speakers. We trained the TTS and ASR models using the real speech of SPS[3]. For the TTS model, a word sequence was converted into 33-class phones. In the ASR task, we used 10k-class BPE units. We tried to adapt the ASR model to the APS subset using the transcription of 151,627 utterances (299.5 hours) as the target domain. For evaluation, we used eval1, which is APS domain speech.

4.2. FastSpeech 2-based TTS and proposed network

We used a FastSpeech 2-based model as the text-to-mel model. The encoder consisted of a 6-layer Transformer block with 384 model dimensions, 1,536 feed-forward network dimensions, and four attention heads. The variance adaptor consisted of three variance predictors which have two CNN layers with a ReLU activation and layer normalization to predict the duration, pitch, and energy. The 6-layer Transformer with 4-head and 384-dimensional hidden states which consumes the output of the variance adaptor predicted 80-dimensional Mel spectrograms with a shift of 12.5 ms. We added the post-net which had five CNN layers with a kernel size 5. For the multi-speaker TTS model, speaker IDs were fed to the encoder and decoder. In speech generation, we randomly selected one speaker ID per one sentence. We used a linear warmup for the 4k steps. The TTS models were trained with a gradient norm clipping of 1.0, and each batch contains totally 10k frames.

The proposed post-processing network consisted of 6-layer Transformer blocks with 384 model dimensions, 1,536 feed-forward network dimensions, and four attention heads. It consumes the predicted Mel spectrogram and the output of variance adaptor. It is trained on the same setting as the FastSpeech 2-based model encoder.

For comparison of our proposed method, we used the VocGAN vocoder [33] which converts Mel spectrograms into a waveform. We implemented it based on an open-source code[4] and trained it using LibriTTS. We changed the up-sampling rates of the generator to 5, 5, 2, 2, and 2 to generate a 16kHz sampling waveform.

Table 1: Results of TED-LIUM 2 dev and test set (WER [%]) and data generation time of the TTS step.

Method	dev	test	time
Baseline model: Real (train-clean-100)	30.19	27.60	–
Adapted Model: Real (train-clean-100) + TTS (TED-LIUM 2)			
w/o vocoder and post-processing	17.12	17.79	1×
w/ vocoder	16.71	16.76	2.75×
Proposed method	**16.71**	**16.04**	**1.26**×
Oracle Model: Real (TED-LIUM 2)	9.28	8.56	–

4.3. Transformer-based ASR system

The ASR model consisted of two CNN subsampling layers (each subsampling factor is 2), 12-layer Conformer-based encoder [5] with 4-head and 256-dimensional hidden states, and 1-layer unidirectional LSTM decoder with an attention mechanism which had 256-dimensional hidden states. We used the 80-dimensional Mel spectrogram with a frame shift of 10ms as the input features for the real speech. In training, we applied a label smoothing [34] with a factor of 0.1, SpecAugment [35], and multi-task learning with the CTC loss. We used a linear warmup for the 25k steps. In the adaptation, we did not try to balance the real and synthetic speech in a batch. In decoding, we set the beam search width to 10. For shallow fusion, we composed a language model with a 4-layer unidirectional LSTM with 512-dimensional hidden states, and the LM weight was set to 0.2.

4.4. Results

Table 1 shows the word error rates (WERs) for TED-LIUM 2 dev and test set and the data generation time relative to that of the FastSpeech 2 model. The data generation time of the model with the vocoder includes conversion of the generated waveform to the Mel spectrograms. When we did not use any generated speech, WERs were not good because there was a serious domain mismatch between LibriSpeech and TED-LIUM 2. By using the generated speech by the TTS model, we observe 43.3% and 35.5% relative improvement on the dev and test set of TED-LIUM 2 without any post-processing. Applying the vocoder yielded further improvement (44.6% and 39.3% relative improvement). Our proposed post-processing network achieved sightly better performance than

[1]We must match only the number of frequency bins. In this work, we used 80-dimensional frequency bins in both ASR and TTS tasks.

[2]https://github.com/Kyubyong/g2p

[3]We used about 160 hours in training the TTS and mel-to-mel models, as training these models with 324.1 hours takes too much time.

[4]https://github.com/rishikksh20/VocGAN

Table 2: Effect of phone information in the proposed method.

	dev	test
w/ phone information (w/ F0 and energy)	**16.71**	**16.04**
w/ phone information (w/o F0 and energy)	16.96	16.30
w/o phone information	17.13	16.76

Table 3: Comparison of frequency bins (80-dim, 0-8kHz) selectively refined in the proposed method. All models used the phone information. Low bin corresponds to low frequency. In "1-20" and "20-80", we did not use any loss weight.

Method	dev	test
1-20	17.20	16.65
21-80	17.21	16.85
1-80	**16.71**	**16.04**

the vocoder (45.3% and 41.9% relative improvement) in a much smaller amount of data generation time. We confirmed that our proposed method enhanced the effect of data generation with a simple framework.

In Table 2, we evaluated the effect of the use of the output of the variance adaptor, which has phone information together with F0 and energy. The model without phone embedding information uses only Mel spectrograms generated by the FastSpeech 2 model. In this case, improvement of the ASR performance is limited and worse than the case using the vocoder. On the other hand, when we remove the additional acoustic prediction (F0 and energy), the result is not changed so much. These results show that the use of the phone embedding information is critical for improving the speech refinement and ASR performance.

Table 3 presents an investigation which frequency bins we should refine. In this experiment, we refined the Mel spectrograms of the specified bins. Enhancing all bins ("1-80") achieved the best performance. Partial refinement improved the performance, but the improvement was limited. Fig. 2 shows the L1 loss of the FastSpeech 2 and proposed model. It indicates that the loss at high frequency bins are larger than that at low frequency bins because the low frequency bins have a constant high energy, which is more easily learned. The generated Mel spectrogram at high frequency bins still has a large gap with the real Mel spectrogram and filling the gap improves the ASR performance. We also confirmed the loss of the proposed model is lower than that of the FastSpeech 2 model in all frequency bins. This suggests the proposed model improves Mel spectrogram effectively.

In the proposed method, we used a residual block and did not predict Mel spectrogram directly. We used a replacement block as an alternative (remove '+' sign in Fig. 1). The network with a replacement block directly predicts Mel spectrogram to be replaced. Table 4 shows that the residual block model achieves higher performance than the replacement block.

Table 5 shows the results of domain adaptation in the Japanese data sets. The model without any processing achieves 40.4% relative improvement from the baseline model. When we compare the augmented and oracle models, the absolute WERs difference is lower than 2 points. This is because the speaking style of SPS is spontaneous and similar to that of APS. The proposed model realizes a large improvement in much less data generation time. In the CSJ experiments, the data generation time of the vocoder is shorter than in TED-LIUM 2 cases since the duration of TED-LIUM 2 speech is longer than that of CSJ (the average duration of TED-LIUM 2 synthesized speech is 7.3s and that of CSJ is 5.3s). We confirm the proposed network refines the synthesized speech effectively in different kinds of data sets.

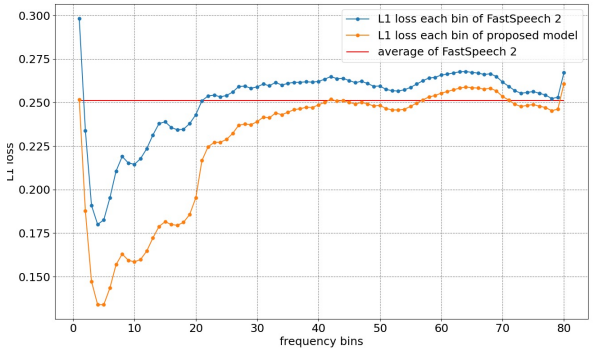

Fig. 2: Values of L1 losses of the FastSpeech 2, proposed model, and the average of frequency bins in the FastSpeech 2. The loss was calculated for each frequency bin from random 1,000 dev-clean samples.

Table 4: Comparison of the residual and replacement blocks in the proposed method. These networks used the phone information and refined all bins.

Method	dev	test
replacement block	17.21	16.97
residual block	**16.71**	**16.04**

Table 5: Results of CSJ test set (WER [%]) and data generation time.

Method	eval1	time
Baseline model: Real (SPS)	17.09	–
Adapted Model: Real (SPS) + TTS (APS)		
w/o vocoder and post-processing	10.19	1×
w/ vocoder	10.09	2.03×
Proposed method	**9.75**	**1.26**×
Oracle Model: Real (SPS+APS)	8.37	–

5. CONCLUSIONS

In this work, we have proposed the phone-informed post-processing network for data augmentation for the ASR model using the TTS model without the vocoder network. Unlike the vocoder network, we directly refine the generated Mel spectrogram derived from the text-to-mel network (FastSpeech 2-based model). The proposed network uses not only the predicted Mel spectrogram but also the output of the variance predictor which corresponds to the phone information. In the experimental evaluations, the proposed method resulted in a large improvement from the baseline and better performance than the vocoder in a much smaller amount of data generation time. We also showed that the use of the phone information is critical for improving the performance.

6. ACKNOWLEDGEMENT

This work was supported in part by JSPS KAKENHI. The authors would like to thank Dr. Shinnosuke Takamichi and Dr. Yuki Saito for their helpful discussions and advices.

7. REFERENCES

[1] Alex Graves, Santiago Fernandez, Faustino Gomez, and Jurgen Schmidhuber, "Connectionist Temporal Classification : Labelling Unsegmented Sequence Data with Recurrent Neural Networks," in *ICML*, 2006, pp. 369–376.

[2] Jan Chorowski and Navdeep Jaitly, "Towards better decoding and language model integration in sequence to sequence models," in *INTERSPEECH*, 2017, pp. 523–527.

[3] Eric Battenberg, Jitong Chen, Rewon Child, Adam Coates, Yashesh Gaur Yi Li, Hairong Liu, Sanjeev Satheesh, Anuroop Sriram, and Zhenyao Zhu, "Exploring neural transducers for end-to-end speech recognition," in *ASRU*, 2017, pp. 206–213.

[4] Linhao Dong, Shuang Xu, and Bo Xu, "Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition," in *ICASSP*, 2018, pp. 5884–5888.

[5] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang, "Conformer: Convolution-augmented Transformer for Speech Recognition," in *INTERSPEECH*, 2020, pp. 5036–5040.

[6] Andrew Rosenberg, Yu Zhang, Bhuvana Ramabhadran, Ye Jia, Pedro Moreno, Yonghui Wu, and Zelin Wu, "Speech recognition with augmented synthesized speech," in *ASRU*, 2019, pp. 996–1002.

[7] Nick Rossenbach, Albert Zeyer, Ralf Schluter, and Hermann Ney, "Generating Synthetic Audio Data for Attention-based Speech Recognition Systems," in *ICASSP*, 2020, pp. 7064–7068.

[8] Zhehuai Chen, Andrew Rosenberg, Yu Zhang, Gary Wang, Bhuvana Ramabhadran, and Pedro J. Moreno, "Improving Speech Recognition Using GAN-Based Speech Synthesis and Contrastive Unspoken Text Selection," in *INTERSPEECH*, 2020, pp. 556–560.

[9] Gary Wang, Andrew Rosenberg, Zhehuai Chen, Yu Zhang, Bhuvana Ramabhadran, Yonghui Wu, and Pedro Moreno, "Improving speech recognition using consistent predictions on synthesized speech," in *ICASSP*, 2020, pp. 7029–7033.

[10] Gary Wang, Andrew Rosenberg, Zhehuai Chen, Yu Zhang, Bhuvana Ramabhadran, and Pedro J. Moreno, "SCADA: Stochastic, Consistent and Adversarial Data Augmentation to Improve ASR," in *INTERSPEECH*, 2020, pp. 2832–2836.

[11] Aleksandr Laptev, Roman Korostik, Aleksey Svischev, Andrei Andrusenko, Ivan Medennikov, and Sergey Rybin, "You do not need more data: Improving end-to-end speech recognition by text-to-speech data augmentation," *International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, 2020.

[12] Amin Fazel, Wei Yang, Yulan Liu, Roberto Barra-Chicote, Yixiong Meng, Roland Maas, and Jasha Droppo, "SynthASR: Unlocking Synthetic Data for Speech Recognition," in *INTERSPEECH*, 2021, pp. 896–900.

[13] Masato Mimura, Sei Ueno, Hirofumi Inaguma, Shinsuke Sakai, and Tatsuya Kawahara, "Leveraging sequence-to-sequence speech synthesis for enhancing acoustic-to-word speech recognition," in *SLT*, 2018, pp. 477–484.

[14] Sei Ueno, Masato Mimura, Shinsuke Sakai, and Tatsuya Kawahara, "Multi-speaker sequence-to-sequence speech synthesis for data augmentation in acoustic-to-word speech recognition," in *ICASSP*, 2019, pp. 6161–6165.

[15] Xianrui Zheng, Yulan Liu, Deniz Gunceler, and Daniel Willett, "Using synthetic audio to improve the recognition of out-of-vocabulary words in end-to-end asr systems," in *ICASSP*, 2021, pp. 5659–5663.

[16] Gakuto Kurata, George Saon, Brian Kingsbury, David Haws, and Zoltán Tüske, "Improving Customization of Neural Transducers by Mitigating Acoustic Mismatch of Synthesized Audio," in *INTERSPEECH*, 2021, pp. 2027–2031.

[17] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, RJ Skerry-Ryan, et al., "Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions," in *INTERSPEECH*, 2017, pp. 4779–4783.

[18] Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, and Ming Liu, "Neural Speech Synthesis with Transformer Network," in *AAAI*, 2019.

[19] Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu, "Fastspeech: Fast, robust and controllable text to speech," in *NeurIPS*, 2019, vol. 32.

[20] Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu, "FastSpeech 2: Fast and high-quality end-to-end text to speech," in *ICRL*, 2020.

[21] Isaac Elias, Heiga Zen, Jonathan Shen, Yu Zhang, Ye Jia, Ron Weiss, and Yonghui Wu, "Parallel tacotron: Non-autoregressive and controllable tts," in *ICASSP*, 2021, pp. 5694–5698.

[22] Isaac Elias, Heiga Zen, Jonathan Shen, Yu Zhang, Ye Jia, RJ Skerry-Ryan, and Yonghui Wu, "Parallel Tacotron 2: A non-autoregressive neural tts model with differentiable duration modeling," in *INTERSPEECH*, 2021, pp. 141–145.

[23] Jean-Marc Valin and Jan Skoglund, "Lpcnet: Improving neural speech synthesis through linear prediction," in *ICASSP*, 2019, pp. 5891–5895.

[24] Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brébisson, Yoshua Bengio, and Aaron C Courville, "MelGAN: Generative adversarial networks for conditional waveform synthesis," in *NeurIPS*, 2019, vol. 32.

[25] Hakan Erdogan, John R. Hershey, Shinji Watanabe, and Jonathan Le Roux, "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks," in *ICASSP*, 2015, pp. 708–712.

[26] Keisuke Kinoshita, Marc Delcroix, Atsunori Ogawa, and Tomohiro Nakatani, "Text-informed speech enhancement with deep neural networks," in *INTERSPEECH*, 2015, pp. 1760–1764.

[27] Kilian Schulze-Forster, Clement S. J. Doire, Gaël Richard, and Roland Badeau, "Joint phoneme alignment and text-informed speech separation on highly corrupted speech," in *ICASSP*, 2020, pp. 7274–7278.

[28] Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J. Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu, "LibriTTS: A corpus derived from librispeech for text-to-speech," *arXiv preprint arXiv:1904.02882*, 2019.

[29] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *ICASSP*, 2015, pp. 5206–5210.

[30] Masanori Morise, Fumiya Yokomori, and Kenji Ozawa, "WORLD: A vocoder-based high-quality speech synthesis system for real-time applications," *IEICE Transactions on Information and Systems*, vol. E99.D, no. 7, pp. 1877–1884, 2016.

[31] Anthony Rousseau, Paul Deléglise, and Yannick Estève, "Enhancing the TED-LIUM corpus with selected data for language modeling and more TED talks," in *LREC*, 2014, pp. 3935–3939.

[32] Kikuo Maekawa, Hanae Koiso, Sadaoki Furui, and Hitoshi Isahara, "Spontaneous Speech Corpus of Japanese," in *LREC*, 2000, pp. 947–9520.

[33] Jinhyeok Yang, Junmo Lee, Youngik Kim, Hoonyoung Cho, and Injung Kim, "VocGAN: A high-fidelity real-time vocoder with a hierarchically-nested adversarial network," in *INTERSPEECH*, 2020, pp. 200–204.

[34] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna, "Rethinking the inception architecture for computer vision," in *CVPR*, 2016, pp. 2818–2826.

[35] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *INTERSPEECH*, 2019, pp. 2613–2617.