Refining Synthesized Speech Using Speaker Information and Phone Masking for Data Augmentation of Speech Recognition

Sei Ueno *Member, IEEE*, Akinobu Lee *Member, IEEE*, Tatsuya Kawahara *Fellow, IEEE*

***Abstract*—While end-to-end automatic speech recognition (ASR) has shown impressive performance, it requires a huge amount of speech and transcription data. The conversion of domain-matched text to speech (TTS) has been investigated as one approach to data augmentation. The quality and diversity of the synthesized speech are critical in this approach. To ensure quality, a neural vocoder is widely used to generate speech waveforms in conventional studies, but it requires a huge amount of computation and another conversion to spectral-domain features such as the log-Mel filterbank (lmfb) output typically used for ASR. In this study, we explore the direct refinement of these features. Unlike conventional speech enhancement, we can use information on the ground-truth phone sequences of the speech and designated speaker to improve the quality and diversity. This process is realized as a Mel-to-Mel network, which can be placed after a text-to-Mel synthesis system such as FastSpeech 2. These two networks can be trained jointly. Moreover, semantic masking is applied to the lmfb features for robust training. Experimental evaluations demonstrate the effect of phone information, speaker information, and semantic masking. For speaker information, x-vector performs better than the simple speaker embedding. The proposed method achieves even better ASR performance with a much shorter computation time than the conventional method using a vocoder.**

***Index Terms*—Speech recognition, Data augmentation, Domain adaptation, Speech synthesis.**

I. INTRODUCTION

RECent findings regarding deep neural networks (DNNs) have led to end-to-end automatic speech recognition (ASR) architectures. While conventional DNN-HMM ASR hybrid systems comprise an acoustic model (AM) and a language model (LM) and train them separately, end-to-end ASR systems integrate the AM and LM and train them using a unified criterion. End-to-end models can recognize faster than the DNN-HMM hybrid models and achieve better performance when a large amount of training data is available. There are several major approaches for end-to-end models: connectionist temporal classification (CTC) [1], [2], [3], [4] and sequence-to-sequence (seq2seq) models such as RNN-transducers [5], attention-based encoder-decoder models [6], [7], [8], [9], [10], [11], and Transformer-based models [12], [13].

Although the integration of AM and LM makes the entire ASR models simple, end-to-end ASR models require a large amount of paired speech and transcription data for training. However, this is not easy to prepare, in particular, for a specific domain. In addition, an ASR system trained in a different domain performs poorly in recognizing speech in the target domain.

Meanwhile, speech-only and text-only data can be more easily available than paired data. Data augmentation and domain adaptation methods using unpaired speech-only or text-only data have been investigated to alleviate data sparseness or domain mismatch. Methods using speech-only data include self-supervised learning (SSL) and semi-supervised approaches. SSL approaches such as wav2vec 2.0 [14] train a powerful representation from speech in the pre-training and then fine-tune on transcribed speech. In semi-supervised learning, we prepare student and teacher ASR models. We make pseudo labels by recognizing the speech using the teacher ASR model. We then re-train the student model using the pseudo-paired data. Although SSL and semi-supervised models achieve high performance, they need a large amount of speech-only data, which takes time for training.

There have also been many works on enhancing the LM function of ASR using text-only data. One of the major approaches is to integrate the ASR model with an external LM, which is trained using a large amount of text-only data. For example, shallow fusion [15], which is widely used in many ASR scenarios, combines the probabilities of an ASR model and an external LM through a log-linear interpolation in beam search decoding. While it is simple yet effective because the external LM is used in only the decoding stage and separately trained, the improvement is limited.

Data augmentation methods that use text-only data with a text-to-speech (TTS) system have also been studied. There are two approaches to using a TTS system. One approach, referred to as speech chain, is to integrate the ASR module and TTS module and train them simultaneously [16], [17]. The integrated modules can be trained using unpaired speech and text data, and the ASR performance is improved in low-resource settings. However, joint optimization could imply that a TTS system generates a speech that the ASR system can easily recognize because the TTS system is jointly trained with the ASR.

Thus, a major approach is to design ASR and TTS independently and generate speech data from a TTS system to make pseudo data. [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29]. In this paper, we focus on this approach. Using this approach, we first train or prepare the TTS system

Manuscript received XXXX, XX, 2023.

S. Ueno and A. Lee are with Nagoya Institute of Technology, Nagoya-shi, Aichi, Japan. e-mail: sei.ueno@nitech.ac.jp, ri@nitech.ac.jp. T. Kawahara is with Graduate School of Informatics, Kyoto University, Sakyo-ku, Kyoto, 606–8501, Japan. e-mail:kawahara@i.kyoto-u.ac.jp

and generate speech from arbitrary text. We then train the ASR model using the synthesized speech. Since we design the ASR and TTS models independently, we can design state-of-the-art ASR and TTS models without changing their architecture. In this approach, the ASR model can effectively learn unknown vocabulary or domains. This approach substantially improves ASR performance, particularly for unknown domains.

However, the ASR improvement is still limited compared with the case where we prepare paired real speech and transcription data. This is because there is still a gap between synthesized speech and real speech. Therefore, we investigate a Mel-to-Mel network that directly refines log-Mel filterbank (lmfb) features generated by the text-to-Mel network [30].

In this paper, we propose a new Mel-to-Mel network to enhance synthesized speech using speaker information and phone masking for further improvement. We add speaker information to the Mel-to-Mel network since this information is generally used in the TTS task to realize multi-speaker TTS systems. Additionally, recent ASR systems introduce masking methods to improve robustness. Inspired by these works, we propose a masking method used when training the Mel-to-Mel network. Finally, we jointly train the text-to-Mel and Mel-to-Mel networks.

Compared with the previous work [30], this paper has three novel contributions.

- We use speaker information in the Mel-to-Mel network to enhance the refinement.
- We introduce a masking method to the output of the text-to-Mel network fed to the Mel-to-Mel network.
- We jointly train the text-to-Mel network with the Mel-to-Mel network.

In the rest of the paper, we first review the text to speech model in Section II. Section III explains the data augmentation approach using TTS for ASR and related works. Section IV describes the proposed method. Experimental settings and evaluations using several settings are presented in Section V and Section VI.

II. Text-to-speech Network

A. Text-to-Mel Network

End-to-end TTS models have recently been investigated as in the case of ASR models because of their simple architecture and fast inference compared with statistical TTS systems. In this section, we review end-to-end TTS models. The TTS model has two separate networks: a text-to-Mel network and a vocoder (Mel-to-waveform network) network. A text-to-Mel network predicts lmfb features from a text sequence. Text is generally converted into a phone or character sequence, and we use phones in this paper.

Recently, non-autoregressive models have been investigated such as FastSpeech [31], [32] and Parallel Tacotron [33], [34]. These networks generate lmfb features faster than autoregressive networks because they can generate all outputs in parallel. The non-autoregressive model is appropriate for data augmentation since we need to generate a large amount of speech. In this paper, we use a FastSpeech 2-based model [32]. The FastSpeech 2 model is composed of a Transformer-based network. The main characteristic of the model is that a network is added, which is called a variance adaptor. The variance adaptor is divided into a duration predictor and optional predictors. The duration predictor predicts the duration of the Mel spectrogram corresponding to each input phone. In addition to the duration, the variance adaptor predicts some prosodic information such as pitch. We implement three predictors: a duration predictor, a pitch predictor, and an energy predictor. Each predictor has a 1-D CNN block + ReLU activation, a layer normalization block, a 1-D CNN block + ReLU activation, a layer normalization block, and a linear layer. The FastSpeech 2 model predicts the duration of the Mel spectrogram with a non-autoregressive architecture. To predict the duration, we prepare the alignment before training the FastSpeech 2-based model.

Formally, let $\mathbf{Y} = (\mathbf{y}_1, ..., \mathbf{y}_L)$ be a length-L input text sequence and $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_T)$ be a length-T output acoustic feature sequence ($L \leq T$). The encoder transforms an input text sequence X into intermediate representation $\mathbf{H} = (\mathbf{h}_1, ..., \mathbf{h}_L)$. The duration predictor predicts the duration of each input text $\mathbf{D} = (d_1, ..., d_L)$, where $d_1 + ... + d_L = T$. We then extend the intermediate features according to $\mathbf{D}$ as follows:

$$\hat{\mathbf{H}} = (\mathbf{h}_1, \mathbf{h}_1, \mathbf{h}_1, ..., \mathbf{h}_L, \mathbf{h}_L) \tag{1}$$

The pitch and energy predictor predicts pitch and energy, respectively, and their predicted acoustic information is embedded. The embedded features are added to the $\hat{\mathbf{H}}$ via residual blocks. Finally, the decoder predicts $\mathbf{X}$ using $\hat{\mathbf{H}}$ in parallel.

In this paper, we use a 5-layer convolutional post-net [35] that predicts a residual to be added to the prediction to improve the overall reconstruction. In training, we minimize five mean absolute error (MAE) losses for the lmfb features of the FastSpeech 2 output and those of the post-net output, duration, pitch, and energy. We need to prepare the alignment, pitch, and energy for training.

B. Vocoder

After generating the lmfb features, we use an additional module to convert them into a waveform, which humans can evaluate. It is referred to as a vocoder. Similar to the text-to-Mel network, vocoder networks are categorized into autoregressive and non-autoregressive models. WaveNet [36] and WaveRNN [37] vocoders are a kind of autoregressive model. They offer tractable likelihood computation, but require an autoregressive generation of the waveform at a much slower inference time.

For faster generation, several works have proposed non-autoregressive models such as Generative Adversarial Network (GAN)-based approaches [38], [39] and diffusion-based approaches [40], [41]. In this paper, we use a GAN-based model. GAN-based vocoders generate a high-quality waveform through faster decoding with fewer parameters. In this paper, we use VocGAN [39].

III. Data Augmentation for ASR Using TTS

In the data augmentation approach used in this paper, we compose the ASR and TTS models individually. In the first stage, we train the text-to-Mel network using the paired data of an lmfb feature and transcription. In the TTS task, we do not have to prepare data where speech and transcription are the same domain as the target for the ASR task. We then generate the lmfb features from arbitrary text.

After generating the speech, we have two ways of augmenting data for ASR. First, we utilize the lmfb features generated by the text-to-Mel network as they are. In this method, the overall generation (inference) is fast since there is no post-processing. However, unnatural speech is often generated. For instance, the non-autoregressive text-to-Mel model generates lmfb features that have an abrupt transition in energy because the non-autoregressive model predicts lmfb feature in parallel. Therefore, as the second method, we use a vocoder network to generate a raw waveform from the generated lmfb feature. We then convert the waveform into lmfb features again and use them as inputs of the ASR models. After generating or converting lmfb features, we mix the real speech with the synthesized speech to train the ASR model.

While many works confirmed that this data augmentation method improves the ASR performance, the improvement is limited compared with using real data. This difference is derived from two aspects: acoustic diversity and speech quality. The acoustic diversity problem is mitigated by introducing speaker embedding to generate multi-speaker speech, but the overall speech quality is degraded.

Several works have attempted to alleviate the problem [18], [23], [24], [27], [26], [22], [28]. Mimura *et al.* [18] enhanced the ASR acoustic encoder in training with synthesized data. Wang *et al.* [23], [24] investigated consistency regularization for TTS incorporated with ASR. Zheng *et al.* [27] introduced a loss for the regularization of the decoder with an ASR model that was fine-tuned for out-of-vocabulary words. Fazel *et al.* [26] investigated a multi-stage training strategy by combining weighted multi-style training, data augmentation, encoder freezing, and parameter regularization. Chen *et al.* [22] introduced a GAN-based model for a pre-trained TTS and ASR to increase the acoustic diversity in the synthesized data. Kurata *et al.* [28] introduced a mapping network before the ASR encoder to convert the acoustic features of the synthesized audio to those of the target domain. The mapping network is added to the ASR architecture and jointly trained with the ASR model. Hu *et al.* [29] introduced a rejection sampling algorithm and separate batch normalization statistics for the real and synthesized features in the ASR stage.

IV. Proposed Method

In this paper, we propose a Mel-to-Mel network to refine lmfb features. It uses speaker information, introduces a masking method, and is jointly trained on a text-to-Mel network. Fig. 1 shows an overview of the proposed method.

A. Mel-to-Mel Network to Refine Lmfb Features

We investigate a method that fills the gap between generated and real speeches with regard to speech quality. We observe that the lmfb features generated by the text-to-Mel network are often unnatural; for instance, there is often an abrupt transition in energy.

In the TTS field, GAN-based postfilter models [42], [43] were introduced to improve the synthesized speech quality. In particular, GAN-based models are mostly used for a vocoder task to realize the high-resolution and mitigate the mismatch between real and synthesized speech. Using a GAN-based vocoder is a straightforward method to alleviate the problem. A vocoder can refine unnatural generation such as abrupt transitions. However, for standard ASR data augmentation, converting into a waveform takes additional time compared with generating lmfb features. Moreover, because the ASR model needs not waveform but lmfb features, we again have to convert a waveform into lmfb features, which requires more time to complete the generation process. Therefore, we design a Mel-to-Mel network that directly refines lmfb features.

We feed the generated lmfb features into the Mel-to-Mel network to directly refine the lmfb features. We can refine the lmfb features without changing the modality, and we can directly use the refined features as input for ASR. This results in a much shorter generation time.

B. Use of Phone Information

In the refining stage, we incorporate phone information as the input of the Mel-to-Mel network to improve the overall performance. For general speech enhancement, masking is widely applied to noisy (not Mel) spectrograms [44], but it cannot use text information because it is not usually available. However, it is well known that enhancement will be improved given the phone information of speech [45], [46], which is available in TTS and data augmentation tasks.

In our seminal work [30], we used phone information to enhance the refinement. To compensate for the length difference between lmfb features and phone sequences, we utilized the output of the variance adaptor in FastSpeech 2.

C. Speaker-informed Mel-to-Mel network

Not only phone information but also speaker information is highly related to the patterns of lmfb features. We also use speaker information in the Mel-to-Mel network to train effectively.

We have options to use speaker information: a speaker ID [47], global style token (GST) [48], and x-vector [49]-based model. We use speaker ID or x-vector-based speaker information on the text-to-Mel network and Mel-to-Mel network since the GST does not always indicate speaker information. For the text-to-Mel network, we use an x-vector to generate multi-speaker speech. When using the x-vector in the Mel-to-Mel network, we utilize a different x-vector from that of the text-to-Mel network, but we randomly select them from the same speakers. In the inference stage, we first decide on the speaker and then randomly use two x-vector features from the same speaker.

D. Training Mel-to-Mel model with phone-wise masked features

In the ASR field, a masking method is widely used to improve the robustness. For instance, SpecAugment [50] masks

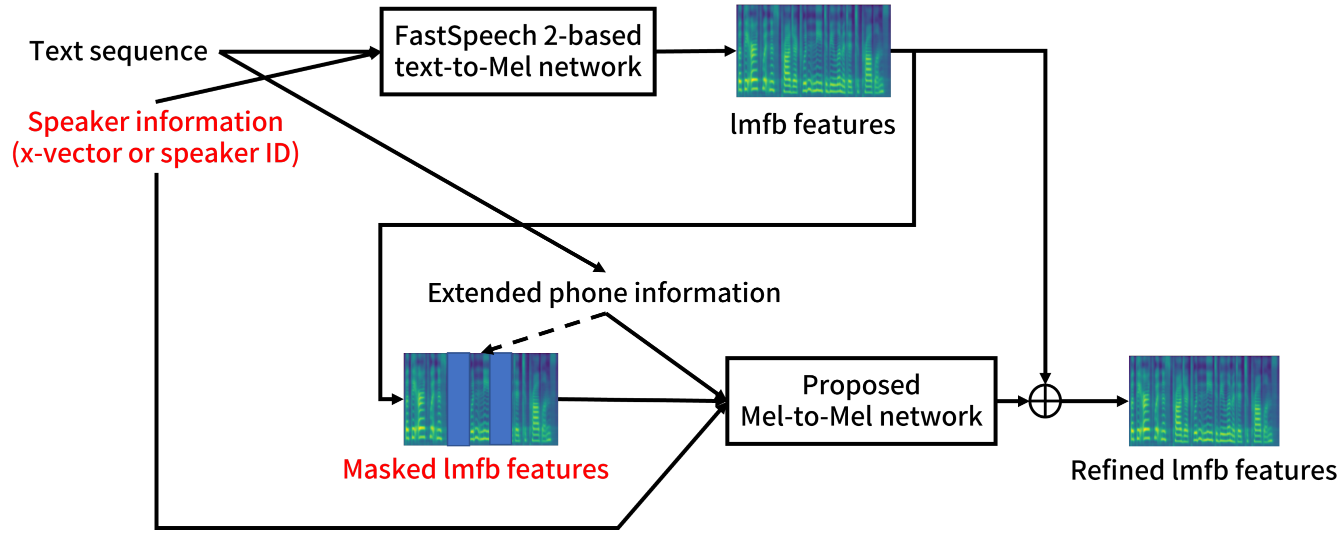
Text sequence
Speaker information (x-vector or speaker ID)
FastSpeech 2-based text-to-Mel network
lmfb features
Extended phone information
Masked lmfb features
Proposed Mel-to-Mel network
Refined lmfb features

Fig. 1. Overview of proposed method. Text (phone) sequence and speaker information (x-vector) are fed into text-to-Mel network. Mel-to-Mel network uses lmfb features and phone information generated by the text-to-Mel network. In training, we also apply a mask on lmfb features corresponding to phone, but masking is not used in inference.

lmfb features for both time and frequency bins and shows effectiveness and consistent improvement. Moreover, in the voice conversion task, Kaneko *et al.* [51] applies a mask to the lmfb features. We apply a mask to lmfb features generated by the text-to-Mel network, and the masked features are fed to the Mel-to-Mel network.

In addition, masking methods that mask lmfb features corresponding to a particular output token (e.g., phones) are more effective than random masking [52]. Thus, we also apply a phone-wise mask to the generated lmfb features.

Formally, let $\mathbf{X}^{gen} = (\mathbf{x}_1^{\text{gen}}, ..., \mathbf{x}_T^{\text{gen}})$ be an lmfb feature sequence generated by the text-to-Mel network. We randomly sample $\mathbf{P} = (p_1, ..., p_L)$ and each $p_l \sim \text{Uniform}(0, 1)$. If $p_l > \sigma$ where σ is a threshold of the masking, the generated acoustic features for all frequency bins corresponding to y_l are masked.

$$\mathbf{x}^{\text{gen\_mask}}_{d_{l-1}^{sum}:d_l^{sum}} = \begin{cases} \langle\text{MASK}\rangle\text{,..., } \langle\text{MASK}\rangle & (p > \sigma) \\ \mathbf{x}^{\text{gen}}_{d_{l-1}^{\text{sum}}:d_l^{\text{sum}}} & (\text{otherwise}) \end{cases}$$

where $\langle\text{MASK}\rangle$ is zero padding and $d_l^{\text{sum}} = \sum_{i=0}^{l} d_i$ (refer to Section IV). Fig. 2 illustrates the inputs of the proposed model. We first generate lmfb features from a text-to-Mel network, which is based on FastSpeech 2. We then mask the generated lmfb features and feed them into a linear layer. We use two linear layers to embed phone information and speaker information and add them to the embedded lmfb feature.

In optimization, we use MAE loss for both masked features and unmasked features. In this paper, we apply a mask to only the generated lmfb features and do not mask the hidden representations for the phone and speaker information since preliminary experiments showed that masking only lmfb features achieves better performance than masking all features.

E. Joint training of text-to-Mel network and Mel-to-Mel network

In the previous work [30], we separately trained text-to-Mel and Mel-to-Mel networks. For separate training, we have two stages. We first train the text-to-Mel network and then fix these parameters. This step is equivalent to a text-to-Mel network generating lmfb features using ground-truth duration, pitch, and energy. Then, we train the Mel-to-Mel network and update only their parameters.

Moreover, we jointly train the text-to-Mel and Mel-to-Mel networks. Recent self-supervised learning (SSL) methods have also introduced masking. They basically have a CNN encoder and Transformer block. The masking is injected between the CNN encoder and Transformer, and they can be jointly trained as a whole network. Inspired by them, when training with masking, we simultaneously optimize the text-to-Mel and Mel-to-Mel networks using their loss functions.

In the joint training, we use a randomly initialized text-to-Mel network and a Mel-to-Mel network[1]. We use five MAE losses for the lmfb features of the text-to-Mel output, duration, pitch, energy, and the proposed Mel-to-Mel network output without any weight.

Note that we do not use post-net when we use the Mel-to-Mel network since the role of post-net is partly the same as the Mel-to-Mel network.

V. Implementation and Experimental Settings

A. Dataset and tasks

We conducted two English domain adaptation experiments. In training the TTS and ASR models in English, we used the LibriTTS [53] and LibriSpeech corpus [54]. LibriTTS (and LibriSpeech) contains read-out speech. LibriTTS is a

[1] When we jointly trained a pre-trained text-to-Mel network and a randomly initialized Mel-to-Mel network, the loss of the whole networks did not decrease compared with the case where we train text-to-Mel and Mel-to-Mel networks from scratch.

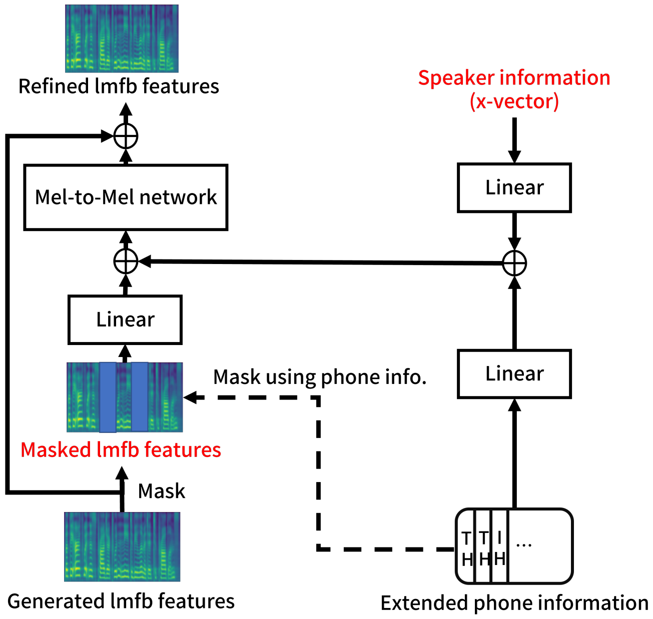

Fig. 2. Use of input features in the proposed method.

sub-corpus of LibriSpeech designed for the TTS task. We downsampled waveforms of LibriTTS to 16kHz to match the sampling rate for all datasets. From LibriTTS and LibriSpeech, we used the paired data in the train-clean-100 subset to train the TTS and ASR models. The train-clean-100 of LibriSpeech contains speech data of 100 hours. The train-clean-100 of LibriTTS contains speech data of 53.8 hours including 246 speakers (Female: 123, Male: 123).

We used two datasets to see the effect of the proposed model. One was the TED-LIUM release-2 (TED-LIUM 2) dataset. TED-LIUM 2 contains spontaneous presentation-style speeches, and it has a 211-hour speech and its transcription. We generated speech from the transcriptions. The other was the LibriSpeech dataset. We generated speech using transcriptions from train-clean-360 and train-other-500. We used only transcription data to generate speech.

B. *FastSpeech 2-based text-to-Mel model*

We composed a 6-layer Transformer block with 384 model dimensions, 1,536 feed-forward network dimensions, and 4 attention heads as the encoder network. The variance adaptor consisted of three variance predictors, which had two CNN layers with a ReLU activation and layer normalization to predict the duration, pitch, and energy. We also composed a 6-layer Transformer with 4-heads, 384 model dimensions, and 1,536 feed-forward network dimensions as the decoder network[2].

For the multi-speaker text-to-Mel system, we used an x-vector [55]. The embedded information of the x-vector feature was added after each self-attention module in all encoder-decoder networks. We used a linear warmup for 4k steps. The TTS models were trained with a gradient norm clipping of 1.0, and each batch contained 10k frames in total. The x-vector was extracted by the pre-trained model, which was trained using Switchboard, Mixer 6, and NIST SREs[3].

To get alignment, we used a CTC-based model. The model had a 5-layer BiLSTM encoder and linear layer. A ground-truth F0 was extracted by WORLD[4] [56].

C. *Mel-to-Mel network model*

The proposed Mel-to-Mel network consisted of a 6-layer Transformer encoder block with 384 model dimensions, 1,536 feed-forward network dimensions, and 4 attention heads. To embed the x-vector and phone information, we added linear layers with the same dimension as the Transformer block. We used a linear warmup for 4k steps. The Mel-to-Mel models were trained with a gradient norm clipping of 1.0, and each batch contained 10k frames in total. In inference, we randomly selected two x-vectors from the same speaker for the text-to-Mel and Mel-to-Mel networks. We used the same x-vector pairs for all settings for a fair comparison. For comparison, we composed StyleGAN 2-based Mel-to-Mel model [57] using open-source code[5]. We used the default settings, and the input of the GAN-based model was lmfb features generated by FastSpeech 2, which is the same as our proposed Mel-to-Mel network.

D. *Vocoder*

For comparing the proposed method, we also implemented a VocGAN vocoder based on an open source[6]. We changed the upsample factor from (4, 4, 2, 2, 2, 2) to (5, 5, 2, 2, 2) to generate 16kHz-sampling waveforms. We used the lmfb features generated by the FastSpeech 2 model as the input of the vocoder to generate a waveform. For vocoder training, we used the generated lmfb features using the text-to-Mel model. When generating lmfb features, the predicted values of duration, F0, and energy were not used, but the ground-truth values extracted from the real waveform data were used.

E. *Conformer-Based ASR Model*

The ASR model consisted of a subsampling network, a Conformer-based encoder network, and a unidirectional LSTM decoder network with an attention mechanism. The subsampling network had two CNN, ReLU activation, and average pooling layers. Each CNN layer had 32 channels with a kernel size of 3×3 and stride factor of 1 with average pooling with a kernel size of 2×2 and stride size of 2×2. The length of the output of the subsampling network was one-quarter of the length of the lmfb features. As the encoder network, we composed a 12-layer Conformer block with 4-heads, 256 model dimensions, 1,536 feed-forward network dimensions, and 31-kernel convolutional layers. The decoder consisted of a 1-layer unidirectional LSTM with an attention mechanism with 256-dimensional hidden states. The ASR model predicted

[2] In the FastSpeech 2 model, the Transformer block of the decoder does not have a cross-attention module.

[3] https://kaldi-asr.org/models/m3

[4] https://github.com/JeremyCCHsu/Python-Wrapper-for-World-Vocoder

[5] https://github.com/NVIDIA/NeMo/blob/main/examples/tts/

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

TABLE I
COMPARISON OF SPEAKER ID VS. X-VECTOR IN TEXT-TO-MEL MODEL FOR TED-LIUM 2 (WER [%])

Method	dev	test
Speaker ID	17.22	16.77
X-vector	16.45	16.40

10k-class BPE-based subwords. We used multi-task learning with a CTC loss for the same subwords. We used a linear warmup for 25k steps. In training the ASR model, we used SpecAugment [50]. We did not mask the generated lmfb features until 25k steps. The ASR model was trained with a gradient norm clipping of 5.0, and each batch contained a total of 50k frames with a gradient accumulation of 4.

In the inference stage, we performed a beam search with a beam size of 10 and a shallow fusion of a language model. To train the LM, we used official unpaired text data of TED-LIUM 2 or LibriSpeech. The LM consisted of a 3-layer unidirectional LSTM with 512 hidden dimensions.

VI. EXPERIMENTAL RESULTS

A. Domain adaptation

First, we compare the speaker information that the TTS model used. TABLE I shows a comparison of methods for TED-LIUM 2 using speaker information for the text-to-Mel model. As seen in this table, we use speaker ID or x-vector for speaker representation. The x-vector was more effective than the speaker ID. This result indicates that the x-vector is appropriate for multi-speaker TTS models since the x-vector represents various factors in not only different speakers but also the same speaker. However, the quality of speaker embedding by x-vector depends on the acoustic differences between the training data used for the x-vector model and that used for speech synthesis. For example, we observed that the loss did not converge when we trained the FastSpeech 2 with an x-vector using the TED-LIUM 2 dataset.

TABLE II shows a comparison of various methods for the TED-LIUM 2 dev and test sets. The WERs of the augmented models were drastically improved because the augmented model efficiently learned acoustic and linguistic patterns in the target domain. Compared with the use of only the text-to-Mel network, the vocoder network yielded 0.31 points in terms of WER improvement on the test set, but the generation time was about 2.3 times as long as the text-to-Mel network. This is because a waveform has much longer sequence lengths than the lengths of lmfb features, and we again need to convert the waveform into lmfb features. The Mel-to-Mel network also improved ASR performance, and the generation time took less than the vocoder model. The GAN-based Mel-to-Mel model achieved almost the same performance as the vocoder.

The Mel-to-Mel network without phone information degraded the ASR performance on the dev set. The Mel-to-Mel network with the x-vector yielded a significant WER improvement compared with the baseline system. Additionally, the masking served to improve the WER. The use of both methods achieved 0.81 and 0.76 points in terms of absolute WER improvement compared with the baseline. Compared with the vocoder and the GAN-based model, the improvement is limited, but the proposed method could keep its faster inference time. The faster generation is a valuable factor because we need to generate a lot of speech samples for ASR training.

Fig. 3 shows an example of lmfb features generated by FastSpeech 2 (top) and our proposed "FastSpeech 2 + Mel-to-Mel network + both (bottom)". We observed that the lmfb features of the proposed method became clear, particularly around 20-40 frequency bins.

TABLE IV, V, VI evaluate the proposed method using metrics that are utilized in the TTS field. In TABLE IV, we used Mel-cepstrum distortion (MCD), root mean square error of F0 (F0 RMSE), and WER. The WER was calculated by the Conformer-LSTM ASR model, which was trained using LibriSpeech 960h. We observed that the MCD and WER did not correlate with any aspects of the performance of the data augmentation. In comparison, the proposed method improved the F0 RMSE. This is because F0 is primarily related to the speaker, and the proposed method can refine the Mel spectrogram by adding speaker information. The proposed method can also consider the F0 transition within the phone by introducing a semantic mask. TABLE V shows mean opinion score (MOS) as a subjective evaluation. In the MOS evaluation, we sampled 200 utterances in dev-clean sets, and 31 native English speakers were asked to make quality judgments about the synthesized speech samples. Each tester listened to 10 samples for each method, for a total of 30 samples. We compared the MOS of the audio samples generated by FastSpeech 2, our proposed method, and real speech. TABLE VI presents a comparison MOS (CMOS) test which compares the proposed method and the baseline method. In the CMOS evaluation, we also sampled 200 different utterances in dev-clean sets, and 35 native English speakers compared the quality. Each tester listened to 15×2 samples. In these results, the proposed method outperformed the baseline FastSpeech 2-based model. The results indicate the proposed method improves speech quality and it can positively affect ASR performance.

TABLE III shows the efficacy of joint training with a mask. Separate training means that we first train the text-to-Mel network and then train the Mel-to-Mel network with fixed parameters of the text-to-Mel network. In comparison, joint training means that we train both text-to-Mel networks and Mel-to-Mel using a unified criterion. We confirmed that joint training with mask and x-vector embedding improved the ASR performance.

Fig. 4 presents an example of refined lmfb features using our proposed method. We observe abrupt transitions in the output of the text-to-Mel network, which are highlighted in red frames. This problem occurs because FastSpeech 2 is a non-autoregressive model. It first predicts the duration of each phone and generates the outputs for the predicted number of frames, which can be abruptly changed in the border of the phone. On the other hand, our proposed method utilizes a Mel-to-Mel transformer that takes into account the context of lmfb features. Additionally, semantic masks encourage prediction using the context, making the method more robust.

TABLE II
RESULTS FOR TED-LIUM 2 (WER [%]). BASELINE MODEL TRAINS ASR TASK USING REAL SPEECH OF TRAIN-CLEAN-100 IN LIBRISPEECH. WE USED SHALLOW FUSION WITH LM TRAINED WITH TED-LIUM 2 OFFICIAL TEXT DATA. * MEANS ONE-SIDED 5% SIGNIFICANT DIFFERENCE, AND ** MEANS TWO-SIDED 5% SIGNIFICANT DIFFERENCE COMPARED WITH BASELINE.

Method	Real [h]	Generated [h]	dev	test	Generation time
w/o any augmentation	100	0	30.19	27.60	—-
Augmented models	100	211			
FastSpeech 2 [baseline]			16.45	16.40	× 1
+ Vocoder			16.26	16.09	× 2.15
+ GAN-based refine			16.27	16.09	× 1.43
+ Mel-to-Mel network w/o phone information			16.84	16.25	× 1.19
+ Mel-to-Mel network [30]			16.17	16.19	× 1.19
+ Mel-to-Mel network (joint training)			16.56	16.43	× 1.19
+ Semantic mask			16.19	*15.85	× 1.19
+ Speaker information (x-vector)			*15.75	*15.88	× 1.19
+ Both			**15.64	**15.64	× **1.19**
Oracle	100 + 211	0	9.28	8.56	—-

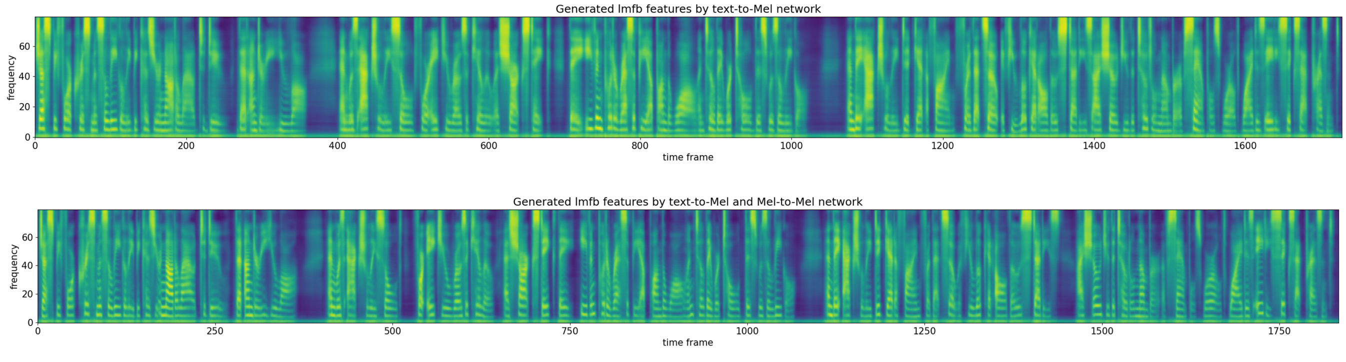

Fig. 3. Example of long lmfb features generated by FastSpeech 2 6-layer (top) and proposed FastSpeech 2 + Mel-to-Mel network w/ x-vector + semantic mask (bottom).

TABLE III
COMPARISON OF JOINT AND SEPARATE TRAINING WITH TED-LIUM 2. IN THIS TABLE, WE ALSO USED SEMANTIC MASK AND X-VECTOR EMBEDDING.

Method	dev	test
Joint training	15.64	15.64
Separate training	16.30	16.09

TABLE IV
RESULTS ON MCD [DB] AND F0 RMSE [HZ], AND WER [%] ON DEV-CLEAN SET.

Method	MCD (↓)	F0 RMSE (↓)	WER (↓)
FastSpeech 2 + vocoder	5.95	85.37	4.85
Proposed method + vocoder	5.96	**84.49**	5.06
Real speech	0.00	0.00	2.95

When comparing with the vocoder, the WER of our proposed method was slightly better. Fig. 5 shows the effects of each method. We found that the lmfb features generated by a text-to-Mel network are blurred and have abrupt transitions. Although the vocoder mitigates the abrupt transition, it does not solve the blurriness. Adding speaker information to the Mel-to-Mel network can make the harmonic structure clear. In contrast to the vocoder and GAN-based refinement methods that do not utilize speaker information, the Mel-to-Mel network proposed in this study can incorporate speaker dependency, resulting in improved speech quality. However, the power transition around 40 frames is unnatural. By introducing a semantic mask, the Mel-to-Mel model can effectively learn the context of lmfb features and dissolve unnatural transitions between phones, but the blurriness around 120 frames remains. When using both methods, blurriness and unnatural transitions are cleared, and overall quality is improved. The proposed method positively affects ASR performance and achieves the best performance by complementing each other's methods.

B. Data augmentation for the same domain

In this section, we conducted data augmentation experiments using the same domain. In this experiment, we augmented LibriSpeech 860h (train-clean-360 and train-other-500) and trained the ASR model using real LibriSpeech 100h and synthesized 860h. TABLE VII shows WERs for the LibriSpeech test sets. We first observe that the augmented models yielded a large WER improvement. We also confirm that each proposed method improved the WER from the standard FastSpeech 2 model. The proposed combination of the semantic mask and speaker information achieved the best performance on dev-clean, dev-other, and test-clean sets. It yielded 0.39, 0.97, 0.32, and 0.77 points in terms of absolute WER improvement compared with the baseline on respective test sets. In dev-clean and test-clean sets, the WER difference between the augmented models and the oracle model is small since they are in the same domain as the training data of the TTS model. In dev-other and test-other sets, there are

TABLE V
MEAN OPINION SCORE (MOS) WITH 95% CONFIDENCE INTERVAL.

Method	MOS
FastSpeech 2 + vocoder	2.80 ± 0.14
Proposed method + vocoder	**3.01** ± 0.14
Real speech	4.17 ± 0.11

TABLE VI
CMOS COMPARISON

Method	CMOS
Proposed method + vocoder	0.00
FastSpeech 2 + vocoder	**-0.13**

larger gaps from the oracle model since these datasets are difficult and different from the train-clean sets. When there is a mismatch between the target speech and the speech data for TTS training, the relative improvement is reduced. For further improvement, we need to consider not only the speaker information but also the recording environment or speaking style in the Mel-to-Mel network.

VII. CONCLUSION

In this paper, we have proposed a Mel-to-Mel network to improve lmfb features generated from a text-to-Mel network. The direct refinement takes less time to generate speech than using a neural vocoder. We used a speaker-informed Mel-to-Mel network to specify the speaker information. In addition to speaker information, we also introduced a masking method for the Mel-to-Mel network. We applied the mask to the lmfb features generated by the text-to-Mel network, and the mask was randomly decided by each phone. In training the Mel-to-Mel network with the mask, we conducted joint training with the text-to-Mel network. In experimental evaluations, the phone information, speaker information, and masking method each resulted in an improvement from the baseline, and the use of all methods further improved the WERs on the TED-LIUM 2 test set. Moreover, we confirmed that the proposed method also yielded improvement on a dataset of the same domain.

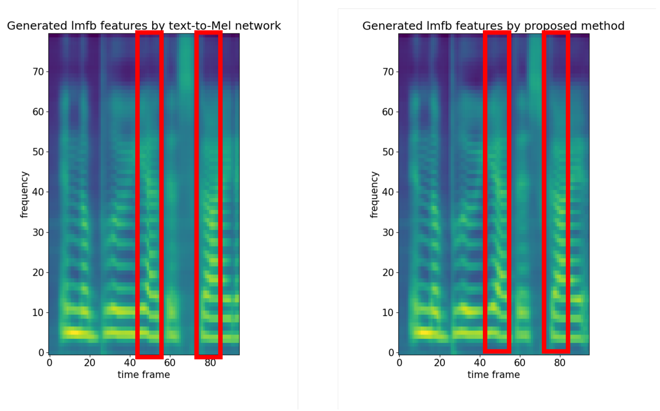

Fig. 4. Example of refinement of our proposed method. The left image shows the lmfb feature generated by the text-to-Mel model, and the right image shows the feature refined by the proposed model. The red frame highlights abrupt transitions, which are smoothed out by the proposed model.

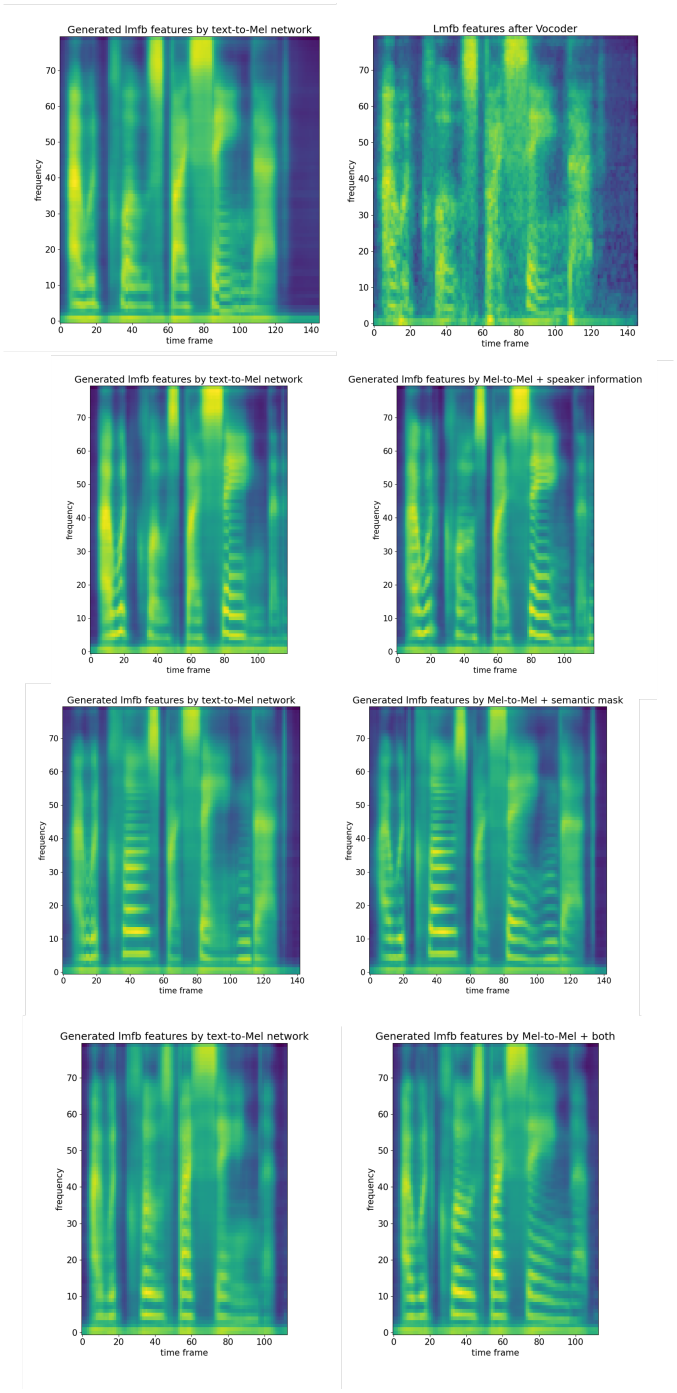

Fig. 5. Comparison of refinement quality of vocoder and our proposed methods. The left image shows the lmfb feature generated by the text-to-Mel model, and the right image shows the refined feature. Note that each text-to-Mel model generates the lmfb feature using the same utterance, but the lmfb features differ due to the text-to-Mel model being trained along with the Mel-to-Mel network and the final parameters being different.

TABLE VII
RESULTS ON WER (%) ON LIBRISPEECH TEST SET. IN THIS EXPERIMENT, WE AUGMENTED SPEECH DATA USING LIBRISPEECH 860H TRANSCRIPTIONS. ** MEANS TWO-SIDED 5% SIGNIFICANT DIFFERENCE, AND *** MEANS TWO-SIDED 1% SIGNIFICANT DIFFERENCE COMPARED WITH BASELINE.

	dev		test	
Method	clean	other	clean	other
w/o any data augmentation	7.13	21.52	7.29	19.92
Augmented model Real 100h + TTS 860h				
FastSpeech 2 [baseline]	4.25	17.35	4.41	16.91
+ Vocoder	4.16	16.54	4.42	16.35
+ GAN-based refine	4.03	16.47	4.47	16.40
+ Mel-to-Mel network w/o phone information	4.16	16.95	4.49	16.68
+ Mel-to-Mel network [30]	4.15	17.06	4.22	15.92
+ Mel-to-Mel network (joint training)	4.11	17.01	4.39	16.57
+ Semantic mask	4.10	***16.57	4.25	***16.05
+ Speaker information (x-vector)	4.15	***16.52	4.23	***16.05
+ Both	***3.86	***16.38	**4.09	***16.14
Oracle (Real 960h)	2.95	7.71	3.00	7.81

ACKNOWLEDGMENT

This work was supported in part by JSPS KAKENHI under Grants JP23K16944.

REFERENCES

[1] A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, “Connectionist Temporal Classification : Labelling Unsegmented Sequence Data with Recurrent Neural Networks,” in *International Conference on Machine Learning (ICML)*, 2006, pp. 369–376.

[2] A. Graves and N. Jaitly, “Towards End-To-End speech recognition with recurrent neural networks,” in *International Conference on Machine Learning (ICML)*, 2014, pp. 1764–1772.

[3] H. Sak, A. Senior, K. Rao, and F. Beaufays, “Fast and Accurate Recurrent Neural Network Acoustic Models for Speech Recognition,” in *INTERSPEECH*, 2015, pp. 1468–1472.

[4] H. Sak, F. de Chaumont Quitry, T. Sainath, and K. Rao, “Acoustic modelling with CD-CTC-SMBR LSTM RNNs,” in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2015, pp. 604–609.

[5] E. Battenberg, J. Chen, R. Child, A. Coates, Y. Gaur Yi Li, H. Liu, S. Satheesh, A. Sriram, and Z. Zhu, “Exploring neural transducers for end-to-end speech recognition,” in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2017, pp. 206–213.

[6] J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio, “End-to-end continuous speech recognition using attention-based recurrent NN: first results,” in *Advances in Neural Information Processing Systems (NIPS) Workshop on Deep Learning*, 2014.

[7] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, “Attention-Based Models for Speech Recognition,” in *Advances in Neural Information Processing Systems (NIPS)*, 2015, pp. 577–585.

[8] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, “Listen, attend and spell: A neural network for large vocabulary conversational speech recognition,” in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2016, pp. 4960–4964.

[9] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, “End-to-End attention-based large vocabulary speech recognition,” in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2016, pp. 4945–4949.

[10] R. Prabhavalkar, T. N. Sainath, B. Li, K. Rao, and N. Jaitly, “An Analysis of “Attention” in Sequence-to-Sequence Models,” in *INTERSPEECH*, 2017, pp. 3702–3706.

[11] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina *et al.*, “State-of-the-art speech recognition with sequence-to-sequence models,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4774–4778.

[12] L. Dong, S. Xu, and B. Xu, “Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition,” in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2018, pp. 5884–5888.

[13] A. Zeyer, P. Bahar, K. Irie, R. Schlüter, and H. Ney, “A comparison of transformer and LSTM encoder decoder models for ASR,” in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 8–15.

[14] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 12 449–12 460.

[15] A. Kannan, Y. Wu, P. Nguyen, T. N. Sainath, Z. Chen, and R. Prabhavalkar, “An analysis of incorporating an external language model into a sequence-to-sequence model,” in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5824–5828.

[16] A. Tjandra, S. Sakti, and S. Nakamura, “Attention-based wav2text with feature transfer learning,” in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2017, pp. 309–315.

[17] ——, “Listening while speaking: Speech chain by deep learning,” in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2017, pp. 301–308.

[18] M. Mimura, S. Ueno, H. Inaguma, S. Sakai, and T. Kawahara, “Leveraging sequence-to-sequence speech synthesis for enhancing acoustic-to-word speech recognition,” in *Workshop on Spoken Language Technology (SLT)*, 2018, pp. 477–484.

[19] S. Ueno, M. Mimura, S. Sakai, and T. Kawahara, “Multi-speaker sequence-to-sequence speech synthesis for data augmentation in acoustic-to-word speech recognition,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6161–6165.

[20] A. Rosenberg, Y. Zhang, B. Ramabhadran, Y. Jia, P. Moreno, Y. Wu, and Z. Wu, “Speech recognition with augmented synthesized speech,” in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 996–1002.

[21] N. Rossenbach, A. Zeyer, R. Schluter, and H. Ney, “Generating Synthetic Audio Data for Attention-based Speech Recognition Systems,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7064–7068.

[22] Z. Chen, A. Rosenberg, Y. Zhang, G. Wang, B. Ramabhadran, and P. J. Moreno, “Improving Speech Recognition Using GAN-Based Speech Synthesis and Contrastive Unspoken Text Selection,” in *INTERSPEECH*, 2020, pp. 556–560.

[23] G. Wang, A. Rosenberg, Z. Chen, Y. Zhang, B. Ramabhadran, Y. Wu, and P. Moreno, “Improving speech recognition using consistent predictions on synthesized speech,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7029–7033.

[24] G. Wang, A. Rosenberg, Z. Chen, Y. Zhang, B. Ramabhadran, and P. J. Moreno, “SCADA: Stochastic, Consistent and Adversarial Data Augmentation to Improve ASR,” in *INTERSPEECH*, 2020, pp. 2832–2836.

[25] A. Laptev, R. Korostik, A. Svischev, A. Andrusenko, I. Medennikov, and S. 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.

[26] A. Fazel, W. Yang, Y. Liu, R. Barra-Chicote, Y. Meng, R. Maas, and J. Droppo, “SynthASR: Unlocking Synthetic Data for Speech Recognition,” in *INTERSPEECH*, 2021, pp. 896–900.

[27] X. Zheng, Y. Liu, D. Gunceler, and D. Willett, “Using synthetic audio to improve the recognition of out-of-vocabulary words in end-to-end asr

systems," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 5659–5663.

[28] G. Kurata, G. Saon, B. Kingsbury, D. Haws, and Z. Tüske, "Improving Customization of Neural Transducers by Mitigating Acoustic Mismatch of Synthesized Audio," in *INTERSPEECH*, 2021, pp. 2027–2031.

[29] T.-Y. Hu, M. Armandpour, A. Shrivastava, J.-H. R. Chang, H. Koppula, and O. Tuzel, "Synt++: Utilizing imperfect synthetic data to improve speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 7682–7686.

[30] S. Ueno and T. Kawahara, "Phone-informed refinement of synthesized mel spectrogram for data augmentation in speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 8572–8576.

[31] 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*, vol. 32, 2019.

[32] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "FastSpeech 2: Fast and high-quality end-to-end text to speech," in *International Conference on Learning Representations (ICRL)*, 2020.

[33] I. Elias, H. Zen, J. Shen, Y. Zhang, Y. Jia, R. Weiss, and Y. Wu, "Parallel tacotron: Non-autoregressive and controllable tts," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 5694–5698.

[34] I. Elias, H. Zen, J. Shen, Y. Zhang, Y. Jia, R. Skerry-Ryan, and Y. Wu, "Parallel Tacotron 2: A non-autoregressive neural tts model with differentiable duration modeling," in *INTERSPEECH*, 2021, pp. 141–145.

[35] 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," in *INTERSPEECH*, 2017, pp. 4779–4783.

[36] 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 *Arxiv*, 2016.

[37] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. van den Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," *CoRR*, vol. abs/1802.08435, 2018.

[38] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brébisson, Y. Bengio, and A. C. Courville, "MelGAN: Generative adversarial networks for conditional waveform synthesis," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[39] J. Yang, J. Lee, Y. Kim, H. Cho, and I. Kim, "VocGAN: A high-fidelity real-time vocoder with a hierarchically-nested adversarial network," in *INTERSPEECH*, 2020, pp. 200–204.

[40] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, and W. Chan, "Wavegrad: Estimating gradients for waveform generation," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=NsMLjcFaO8O

[41] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "Diffwave: A versatile diffusion model for audio synthesis," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=a-xFK8Ymz5J

[42] T. Kaneko, S. Takaki, H. Kameoka, and J. Yamagishi, "Generative adversarial network-based postfilter for stft spectrograms," in *INTERSPEECH*, 2017.

[43] L. Sheng and E. N. Pavlovskiy, "Reducing over-smoothness in speech synthesis using generative adversarial networks," *International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON)*, pp. 0972–0974, 2018.

[44] H. Erdogan, J. R. Hershey, S. Watanabe, and J. Le Roux, "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 708–712.

[45] K. Kinoshita, M. Delcroix, A. Ogawa, and T. Nakatani, "Text-informed speech enhancement with deep neural networks," in *INTERSPEECH*, 2015, pp. 1760–1764.

[46] K. Schulze-Forster, C. S. J. Doire, G. Richard, and R. Badeau, "Joint phoneme alignment and text-informed speech separation on highly corrupted speech," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7274–7278.

[47] A. Gibiansky, S. Arik, G. Diamos, J. Miller, K. Peng, W. Ping, J. Raiman, and Y. Zhou, "Deep voice 2: Multi-speaker neural text-to-speech," in *Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 2962–2970.

[48] Y. Wang, D. Stanton, Y. Zhang, R.-S. Ryan, E. Battenberg, J. Shor, Y. Xiao, Y. Jia, F. Ren, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5180–5189.

[49] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5329–5333.

[50] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *INTERSPEECH*, 2019, pp. 2613–2617.

[51] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "Maskcyclegan-vc: Learning non-parallel voice conversion with filling in frames," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[52] C. Wang, Y. Wu, Y. Du, J. Li, S. Liu, L. Lu, S. Ren, G. Ye, S. Zhao, and M. Zhou, "Semantic Mask for Transformer based End-to-End Speech Recognition," in *INTERSPEECH*, 2020, pp. 971–975.

[53] H. Zen, V. Dang, R. Clark, Y. Zhang, R. J. Weiss, Y. Jia, Z. Chen, and Y. Wu, "LibriTTS: A corpus derived from librispeech for text-to-speech," *arXiv preprint arXiv:1904.02882*, 2019.

[54] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[55] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5329–5333, 2018.

[56] M. Morise, F. Yokomori, and K. 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.

[57] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 8107–8116.