Abstract

With rapid progress in neural text-to-speech (TTS) models, personalized speech generation is now in high demand for many applications. For practical applicability, a TTS model should generate high-quality speech with only a few audio samples from the given speaker, that are also short in length. However, existing methods either require to fine-tune the model or achieve low adaptation quality without fine-tuning. In this work, we propose StyleSpeech, a new TTS model which not only synthesizes high-quality speech but also effectively adapts to new speakers. Specifically, we propose Style-Adaptive Layer Normalization (SALN) which aligns gain and bias of the text input according to the style extracted from a reference speech audio. With SALN, our model effectively synthesizes speech in the style of the target speaker even from a single speech audio. Furthermore, to enhance StyleSpeech’s adaptation to speech from new speakers, we extend it to Meta-StyleSpeech by introducing two discriminators trained with style prototypes, and performing episodic training. The experimental results show that our models generate high-quality speech which accurately follows the speaker’s voice with single short-duration (1-3 sec) speech audio, significantly outperforming baselines.

1. Introduction

In the past few years, the fidelity and intelligibility of speech produced by neural text-to-speech (TTS) synthesis models have shown dramatic improvements. Furthermore, a number of applications such as AI voice assistant services and audio navigation systems have been actively developed and deployed to real-world, attracting increasing demand of TTS synthesis. The majority of the TTS models aim to synthesize high quality speech of a single speaker from the given text (Oord et al., 2016; Wang et al., 2017; Shen et al., 2018; Ren et al., 2019; 2020) and have been extended to support multi speakers (Gibiansky et al., 2017; Ping et al., 2017; Chen et al., 2020).

Meanwhile, there is an increasing demand for personalized speech generation, which requires TTS models to generate high-quality speech that well captures the voice of the given speaker with only a few samples of the speech data. However, natural human speech is highly expressive and contains rich information, including various factors such as the speaker identity and prosody. Thus, generating personalized speech from a few speech audios, potentially even from a single audio, is an extremely challenging task. To achieve this goal, the TTS model should be able to generate speech of multiple speakers and adapt well to an unseen speaker’s voice.

A popular approach to handle this challenge is to pre-train the model on a large dataset consisting of the speech from many speakers and fine-tune the model with a few audio samples of a target speaker (Chen et al., 2019; Arik et al., 2018; Chen et al., 2021). However, this approach requires audio samples and corresponding transcripts of the target speaker to fine-tune, as well as hundreds of fine-tuning steps, which limits its applicability to real-world scenarios. Another approach is to use a piece of reference speech audio to extract a latent vector that captures the voice of the speaker such as speaker identity, prosody and speaking style (Skerry-Ryan et al., 2018; Wang et al., 2018; Jia et al., 2018). Specifically, these models are trained to synthesize speech conditioned on a latent style vector extracted from the speech of the given speaker, in addition to text input. This style-based approach has shown convincing results in expressive speech generation, and is able to adapt to new speakers without fine-tuning. However, they heavily rely on the diversity of the speakers in the source dataset, and thus often show low adaptation performance on new speakers.

Meta-learning (Thrun & Pratt, 1998), or learning to learn, has attracted attention of many researchers recently, as it allows the trained model to rapidly adapt to new tasks only with a few examples. In this regard, some of the few-shot generative models have utilized meta-learning for improved
Meta-StyleSpeech: Multi-Speaker Adaptive Text-to-Speech Generation

generalization (Rezende et al., 2016; Bartunov & Vetrov, 2018; Clouâtre & Demers, 2019). Closely related to few-shot generation, few-shot classification also is the most extensively studied problem for meta-learning. In particular, metric-based methods (Snell et al., 2017; Misra, 2020), which meta-learn a space where the instances from the same class are embedded closer while instances that belong to different classes are embedded farther apart, have shown to achieve high performance. However, the existing methods on few-shot generation and classification mostly targets image domains, and are not straightforwardly applicable to TTS.

To overcome these difficulties, we propose StyleSpeech, a high-quality and expressive multi-speaker adaptive text-to-speech generation model. Our model is inspired by Karras et al. (2019) proposed for image generation, that are shown to generate surprisingly realistic photos of human faces. Specifically, we propose Style-Adaptive Layer Normalization (SALN) which aligns gain and bias of the text input according to a style vector extracted from a reference speech audio. Additionally, we further propose Meta-StyleSpeech, which is meta-learned with discriminators to further improve the model’s ability to adapt to new speakers that have not been seen during training. In particular, we perform episodic training by simulating the one-shot adaptation case in each episode, while additionally training two discriminators, a style discriminator and a phoneme discriminator with adversarial loss. Furthermore, the style discriminator learns a set of style prototypes enforcing the generator to generate speech from each speaker to be embedded closer to its correct style prototype which can be a voice identity of the speaker.

Our main contributions are as follows:

• We propose StyleSpeech, a high-quality and expressive multi-speaker adaptive TTS model, which can flexibly synthesize speech with the style extracted from a single short-length reference speech audio.

• We extend StyleSpeech to Meta-StyleSpeech which adapts well to speech from unseen speakers, by introducing the phoneme and style discriminators with style prototypes and an episodic meta-learning algorithm

• Our proposed models achieve start-of-the-art TTS performance across multiple tasks, including multi-speaker speech generation and one-shot short-length speaker adaptation.

2. Related Work

Text-to-Speech Neural TTS models have shown a rapid progress, including WaveNet (Oord et al., 2016), DeepVoice1, 2, 3 (Arik et al., 2017; Gibiansky et al., 2017; Ping et al., 2017), Char2Wav (Sotelo et al., 2017) and Tacotron1, 2 (Wang et al., 2017; Shen et al., 2018). These models mostly resort to autoregressive generation of mel-spectrogram, which suffers from slow inference speed and a lack of robustness (word missing and skipping). Recently, several works such as Paranet (Peng et al., 2020) and FastSpeech1 (Ren et al., 2019) have proposed non-autoregressive TTS models to handle such issues and achieve fast inference speed and improved robustness over autoregressive models. Besides, FastSpeech2 (Ren et al., 2020) extend FastSpeech1 by using additional pre-obtained acoustic features such as pitch and energy so that they show more expressive speech generation. However, these models only support a single speaker system. In this work, we base our model on FastSpeech2 and we propose an additional component to generate various voice of multi speakers.

Speaker Adaptation As the demand for personalized speech generation have increased, adaptation of the TTS models to new speakers has been extensively studied. A popular approach is to train the model on a large multi-speakers dataset and then fine-tune the whole model (Chen et al., 2019; Arik et al., 2018) or only parts of the model (Moss et al., 2020; Zhang et al., 2020; Chen et al., 2021). As an alternative approach, some of recent works attempted to model the style directly from a speech audio sample. For example, Nachmani et al. (2018) extend VoiceLoop (Taigman et al., 2018) with an additional speaker encoder. Tacotron-based approaches (Skerry-Ryan et al., 2018; Wang et al., 2018) use a piece of reference speech audio to extract the style vector and synthesize speech with the style vector, in addition to text input. Moreover, GMVAE-Tacotron (Hsu et al., 2019) present a variational approach with Gaussian Mixture prior in style modeling. To certain extent, these methods appear to be able to generate speech, that are not limited to trained speakers. However, they often achieve low adaptation performance on unseen speakers, especially when the reference speech is short in length. To handle this issue, we introduce a novel method to flexibly synthesize speech with the style vector and propose meta-learning to improve adaptation performance of our model on unseen speakers.

Meta Learning In recent years, diverse meta-learning algorithms have been proposed, mostly focusing on the few-shot classification problem. Among them, metric-based meta-learning (Vinyals et al., 2016; Snell et al., 2017; Ore-shkin et al., 2018) aims to learn the embedding space where the instances of the same class are embedded closer, while the instances belonging to different classes are embedded farther apart. On the other hand, some of existing works have studied the problem of few-shot generation. For one-shot image generalization, Rezende et al. (2016) propose sequential generative models that are built on the principles of
Meta-StyleSpeech: Multi-Speaker Adaptive Text-to-Speech Generation

3. StyleSpeech

In this section, we first describe the architecture of StyleSpeech for multi-speaker speech generation. StyleSpeech is comprised of a mel-style encoder and a generator. The overall architecture of StyleSpeech is shown in Figure 1.

3.1. Mel Style Encoder

The mel-style encoder, $E_{\text{Enc}_s}$, takes a reference speech $X$ as input. The goal of the mel-style encoder is to extract a vector $w \in \mathbb{R}^N$ which contains the style such as speaker identity and prosody of given speech $X$. Similar to Arik et al. (2018), we design the mel-style encoder to comprise of the following three parts:

1) Spectral processing: We first input the mel-spectrogram into fully-connected layers to transform each frames of mel-spectrogram into hidden sequences.

2) Temporal processing: We then use gated CNNs (Dauphin et al., 2017) with residual connection to capture the sequential information from the given speech.

3) Multi-head self-attention: Then we apply a multi-head self-attention with residual connection to encode the global information. In contrast to Arik et al. (2018) where the multi-head self attention is applied across audio samples, we apply it at the frame level so that the mel-style encoder can better extract style information even from a short speech sample. Then, we temporally average the output of the self-attention to get a one-dimensional style vector $w$.

3.2. Generator

The generator, $G$, aims to generate a speech $\tilde{X}$ given a phoneme (or text) sequence $t$ and a style vector $w$. We build the base generator architecture upon FastSpeech2 (Ren et al., 2020), which is one of the most popular single-speaker models in non-autoregressive TTS. The model consists of three parts; a phoneme encoder, a mel-spectrogram decoder and a variance adaptor. The phoneme encoder converts a sequence of phoneme embedding into a hidden phoneme sequence. Then, the variance adaptor predicts different variances in the speech such as pitch and energy in phoneme-level.

Furthermore, the variance adaptor predict a duration of each phonemes to regulate the length of the hidden phoneme sequence into the length of speech frames. Finally, the mel-spectrogram decoder converts the length-regulated phoneme hidden sequence into mel-spectrogram sequence. Both the phoneme encoder and mel-spectrogram decoder are composed of Feed-Forward Transformer blocks (FFT blocks) based on the Transformer (Vaswani et al., 2017) architecture. However, this model does not generate speech with diverse speakers, and thus we propose a novel component to support multi-speaker speech generation, in the following paragraph.

Style-Adaptive Layer Norm

Conventionally, the style vector is provided to the generator simply through either the concatenation or the summation with the encoder output or the decoder input. In contrast, we apply an alternative approach by proposing the Style-Adaptive Layer Normalization (SALN). SALN receives the style vector $w$ and predicts the gain and bias of the input feature vector. More precisely, given feature vector $h = (h_1, h_2, \ldots, h_H)$ where $H$ is the dimensionality of the vector, we derive the normalized vector $y = (y_1, y_2, \ldots, y_H)$ as follows:

$$y = \frac{h - \mu}{\sigma}$$

$$\mu = \frac{1}{H} \sum_{i=1}^{H} h_i, \quad \sigma = \sqrt{\frac{1}{H} \sum_{i=1}^{H} (h_i - \mu)^2}$$

In multi-style generation, we find that it is straightforward to predict the phoneme-level information than speech frame-level information as in FastSpeech2.
Then, we compute the gain and bias with respect to the style vector \( w \):

\[
SALN(h, w) = g(w) \cdot y + b(w) \tag{2}
\]

Unlike the fixed gain and bias as in LayerNorm (Ba et al., 2016), \( g(w) \) and \( b(w) \) can adaptively perform scaling and shifting of the normalized input features based on the style vector. We substitute SALN for layer normalizations in FFT blocks in the phoneme encoder and the mel-spectrogram decoder. The affine layer which convert the style vector into bias and gain is a single fully connected layer. By utilizing SALN, the generator can synthesize various styles of speech of multiple speakers given the reference audio sample in addition to the phoneme input.

3.3. Training

In the training process, both the generator and the mel-style encoder are optimized by minimizing a reconstruction loss between a mel-spectrogram synthesized by the generator and a ground truth mel-spectrogram\(^2\). We use the \( L_1 \) distance as a loss function, as follows:

\[
\tilde{X} = G(t, w) \quad w = Enc_s(X) \tag{3}
\]

\[
\mathcal{L}_{recon} = \mathbb{E} \left[ \| \tilde{X} - X \|_1 \right] \tag{4}
\]

\(^2\)The reconstruction loss includes pitch, energy, and duration losses as in FastSpeech2, however, we do not present them for brevity.
4.1. Discriminators

The style discriminator, $D_s$, predicts whether the speech follows the voice of the target speaker. The discriminator has similar architecture with mel-style encoder except it contains a set of style prototypes $S = \{s_i\}_{i=1}^{K}$, where $s_i \in \mathbb{R}^N$ denotes the style prototype for the $i$th speaker and $K$ is the number of speakers in the training set. Given the style vector, $w_s \in \mathbb{R}^N$, as input, the style prototype $s_i$ is learned with following classification loss.

$$L_{cls} = -\log \frac{\exp(w_s^T s_i)}{\sum_{s_i} \exp(w_s^T s_i)}$$

(5)

In detail, the dot product between the style vector and all style prototypes is computed to produce style logits, followed by cross entropy loss that encourages the style prototype to represent the target speaker’s common style such as speaker identity.

The style discriminator then maps the generated speech $\tilde{X}_q$ to a $M$-dimensional vector $h(\tilde{X}_q) \in \mathbb{R}^M$ and compute a single scalar with the style prototype. The key idea here is to enforce the generated speech to be gathered around the style prototype for each speaker. In other words, the generator learns how to synthesize speech that follows the common style of the target speaker from a single short reference speech sample. Similar to the idea of Miyato & Koyama (2018), the output of the style discriminator is then computed as:

$$D_s(\tilde{X}_q, s_i) = w_0 s_i^T V h(\tilde{X}_q) + b_0$$

(6)

where $V \in \mathbb{R}^{N \times M}$ is a linear layer and $w_0$ and $b_0$ are learnable parameters. The discriminator loss function of $D_s$ then becomes

$$L_{D_s} = \mathbb{E}_{t,w,s_i \sim S}[(D_s(X_s, s_i) - 1)^2 + D_s(\tilde{X}_q, s_i)^2]$$

(7)

The discriminator loss follows LS-GAN (Mao et al., 2017), which replace the binary cross-entropy terms of the original GAN (Goodfellow et al., 2014) objective with least squares loss functions.

The phoneme discriminator, $D_t$, takes $\tilde{X}_q$ and $t_q$ as inputs, and distinguishes the real speech from the generated speech given the phoneme sequence $t_q$ as the condition. In particular, the phoneme discriminator consists of fully-connected layers and is applied at frame level. Since we know the duration of each phoneme, we can concatenate each frame in mel-spectrogram with the corresponding phoneme. The discriminator then computes scalars for each frame and averages them to get a single scalar. The final discriminator loss function for $D_t$ then is given as:

$$L_{D_t} = \mathbb{E}_{t,w}[(D_t(X_s, t_s) - 1)^2 + D_t(\tilde{X}_q, t_q)^2]$$

(8)

Finally, the generator loss for query speech can be defined as the sum of the adversarial loss for each discriminator, as follows:

$$L_{adv} = \mathbb{E}_{t,w,s_i \sim S}[(D_s(G(t_q, w_s), s_i) - 1)^2 + \mathbb{E}_{t,w}((D_t(G(t_q, w_s), t_q) - 1)^2].$$

(9)

Furthermore, we additionally apply a reconstruction loss for the support speech, as we empirically find that it improves the quality of the generated mel-spectrograms.

$$L_{recon} = \mathbb{E} [||G(t_s, w_s) - X_s||_1]$$

(10)

4.2. Episodic meta-learning

Overall, we conduct the meta-learning of Meta-StyleSpeech by alternating between the updates of the generator and mel-style encoder that minimizes $L_{recon}$ and $L_{adv}$ losses, and the updates of the discriminators that minimizes $L_{D_s}, L_{D_t}$, and $L_{cls}$ losses. Thus, the final meta-training loss to minimize is defined as:

$$L_G = \alpha L_{recon} + L_{adv}$$

(11)

$$L_D = L_{D_s} + L_{D_t} + L_{cls}$$

(12)

We set $\alpha = 10$ in our experiments.

5. Experiment

In this section, we evaluate the effectiveness of Meta-StyleSpeech and StyleSpeech on few-shot text-to-speech synthesis tasks. The audio samples are available at https://stylespeech.github.io/.

5.1. Experimental Setup

Datasets We train StyleSpeech and Meta-StyleSpeech on LibriTTS dataset (Zen et al., 2019), which is a multi-speaker English corpus derived from LibriSpeech (Panayotov et al., 2015). LibriTTS contains 110 hours audios of 1141 speakers and their corresponding text transcripts. We split the dataset into a training and a validation (test) set, and use the validation set for the evaluation on the trained speakers. For evaluation of the models’ performance on unseen speaker adaptation tasks, we use the VCTK (Yamagishi et al., 2019) dataset which contains audios of 108 speakers.

Preprocessing We convert the text sequences into the phoneme sequences with an open-source grapheme-to-phoneme tool\(^3\) and take the phoneme sequences as input. We downsampled an audio to 16kHz and trimmed leading and trailing silence using Librosa (McFee et al., 2015). We

\(^3\)https://github.com/Kyubyong/g2p
extract a spectrogram with a FFT size of 1024, hop size of 256, and window size of 1024 samples. Then, we convert it to a mel-spectrogram with 80 frequency bins. In addition, we average ground-truth pitch and energy by duration to get phoneme-level pitch and energy.

**Implementation Details** The generator in StyleSpeech uses 4 FFT blocks on both phoneme encoder and mel-spectrogram decoder following FastSpeech2. In addition, we add pre-nets to the phoneme encoder and the mel-spectrogram decoder. In detail, the encoder pre-net consists of two convolution layers and a linear layer with residual connection and the decoder pre-net consists of two linear layers. The architecture of pitch, energy and duration predictor in the variance adaptor are the same as those of FastSpeech2. However, instead of using 256 bins for pitch and energy, we use a 1D convolution layer to add real or predicted pitch and energy directly to the phoneme encoder output (La’ncucki, 2020). For the mel-style encoder, the dimensionality of all latent hidden vectors are set to 128, including the size of the style vector. Furthermore, we use Mish (Misra, 2020) activation for both the generator and the mel-style encoder.

The style discriminator has the similar architecture as the mel-style encoder, but we use 1D convolutions instead of gated CNNs. Specifically, the mel-spectrogram passes through two fully connected layers with 256 hidden dimensions before concatenating with phoneme embeddings, and then through three fully connected layers with 512 hidden dimensions and one final projection layer. We use LeakyReLU activation for both discriminators. In addition, we apply spectral normalization (Miyato et al., 2018) in all the layers for both discriminators except style prototypes. The more details about the architectures are in the supplementary material.

In the training process, we train StyleSpeech for 100k steps. For Meta-StyleSpeech, we start from pretrained StyleSpeech that is trained for 60k steps, and then meta-train the model for additional 40k steps via episodic training. We find that meta training from pretrained StyleSpeech helps obtain better stability in training. Furthermore, we use the predicted pitch and energy from the variance adaptor to generate both support and query speech in Meta-StyleSpeech training. In addition, we train our models with a minibatch size of 48 using support and query speech in Meta-StyleSpeech training. In Meta-StyleSpeech, we start from pretrained StyleSpeech for additional 40k steps via episodic training. We find that meta training from pretrained StyleSpeech helps obtain better stability in training.

**Baselines** We compare our model with several baselines. 1) **GT (oracle):** This is the Ground-Truth speech. 2) **GT mel (oracle):** This is the speech synthesized by MelGAN vocoder using Ground-Truth mel-spectrogram. 3) **DeepVoice3** (Ping et al., 2017): This is a multi-speaker TTS model which learns a look-up table to map embeddings for different speaker identity. Since DeepVoice3 is able to generate only seen speakers, we only compare against DeepVoice3 on seen speaker evaluation task. 4) **GMVAE** (Hsu et al., 2019): This is a multi-speaker TTS model based on the Tacotron with a variational approach using Gaussian Mixture prior. 5) **Multi-speaker FS2 (vanilla):** This is a multi-speaker FastSpeech2, which adds the style vector to the encoder output and the decoder input. The style vector is extracted by the mel-style encoder. 6) **Multi-speaker FS2 + d-vector:** This is same as 5) **Multi-speaker FS2 (vanilla)** except that the style vector is extracted from a pre-trained speaker verification model as suggested in Jia et al. (2018). 7) **StyleSpeech:** This is our proposed model which generates multi-speaker speech from a single speech audio with the style-adaptive layer normalization and the mel-style encoder. 8) **Meta-StyleSpeech:** This is our proposed model which extends StyleSpeech to perform meta training, with
**Evaluation Metric** The evaluation of the TTS models is very challenging, due to its subjective nature in the evaluation of the perceptual quality of generated speech. However, we use a sufficient number of metrics to evaluate the performance of our model. Specifically, for subjective evaluation, we conduct human evaluations with **MOS** (mean opinion score) for naturalness and **SMOS** (similarity MOS) for similarity. Both metrics are rated in 1-to-5 scale and reported with the 95% confidence intervals (CI). 50 judges were participated in each experiment, where they were allowed to evaluate each audio sample once, that are presented in random orders.

In addition to subjective evaluation, we also conduct objective evaluation with quantitative measures. **MCD** evaluates the compatibility between the spectra of two audio sequences. Since the sequences are not aligned, we perform Dynamic Time Warping to align the sequences prior to comparison. In addition, **WER** validates an intelligibility of generated speech. We use a pre-trained ASR model DeepSpeech2 (Amodei et al., 2016) to compute Word Error Rate (WER). Note that both MCD and WER are not absolute metrics for evaluating speech quality, so we only use them for relative comparisons.

Beyond the quality evaluation, we also evaluate how similar the generated speech is, to the voice of the target speaker. In particular, we use the speaker verification model based on Wan et al. (2018) to extract x-vectors from generated speech as well as the actual speech of the target speaker. We then compute cosine similarity between them and denote it as **Sim**. The Sim scores are between -1 and 1, with higher scores indicating higher similarity. Since the verification model can be used for any speakers, we use it to evaluate both the trained speakers and new speakers. Furthermore, we use a speaker classifier that identifies which speaker an audio sample belongs to. We train the classifier on VCTK dataset and the classifier achieve 99% accuracy for validation set. We then calculate its **accuracy** to conduct the evaluation for adaptability of our model to new speakers. Better adaptation would result in higher classification accuracy.

### 5.2. Evaluation on Trained Speakers

Before investigating the ability of our model on unseen speaker adaptation, we first evaluate the quality of speech...
Meta-StyleSpeech : Multi-Speaker Adaptive Text-to-Speech Generation

5.3. Unseen Speaker Adaptation

We now validate the adaptation performance of our model on unseen speakers. To this end, we first evaluate the quality of generated speech of unseen speakers. In this experiment, we also randomly draw one audio sample as reference for each 108 unseen speaker in the VCTK datasets and then generate speech using the reference speech and the given text. Table 3 shows the results of MOS, MCD and WER evaluation on speech from unseen speakers. Meta-StyleSpeech also achieves the best generation quality in all three metrics, largely outperforming the baselines.

As done in the seen-speaker experiments, we also evaluate the similarity to reference speech for unseen speakers. Following Nachman et al. (2018), we expect that the ability to adapt to new speakers may depend on the length of the reference speech audio. Thus, we perform the experiments while varying the length of the reference speech audio from unseen speakers. We use four different lengths: < 1 sec, 1 ~ 3 sec, 1 sentence and 2 sentences. We set 1 sentence as a speech audio sample which is longer than 3 seconds in length and, for 2 sentences, we simply concatenate two speech audio samples.

The results of SMOS, Sim, and Accuracy are presented in Table 4. As shown in the result, our model, Meta-StyleSpeech, significantly outperforms the baseline as well as StyleSpeech on reference speech of any lengths. Specifically, it achieves high adaptation performance even with the reference speech that is shorter than 1 second. Figure 4 shows the example of generated speech from the reference speech shorter than 1 second, using Meta-StyleSpeech. We observe that our model generates high quality speech with sharp harmonics and well resolved formants which is comparable to ground-truth mel-spectrogram.

In addition, we also conduct adaptation evaluation depends on various styles such as gender and accent. The result is shown in Table 5. We can see that Meta-StyleSpeech shows balanced results for gender. Moreover, the model also shows high adaptation performance for all accents. However, we can see that slightly lower performance on Indian accent. This could be because the Indian accent have more dramatic variation in speaking style than other accents.

Visualization of style vector To better understand effectiveness of meta-learning, we visualize the style vectors. In figure 3, we demonstrate the t-SNE projection (Maaten & Hinton, 2008) of style vectors from unseen speakers from the VCTK datasets. In particular, we select 10 female speakers who have similar accents and voices which are difficult to distinguish by the model. We can see that while
StyleSpeech clearly better separates the style vectors when compared with GMVAE. Meta-StyleSpeech trained with meta-learning achieves even better clustered style vectors.

5.4. Ablation Study

We further conduct an ablation study to verify the effectiveness of each component in our model, including the text discriminator, the style discriminator and the style prototypes. In the ablation study, we use two metrics; MCD and Accuracy. The results are shown in Table 6. Both the quality of generated speech and adaptation ability of the model are significantly dropped when removing the text discriminator. This indicates that the role of the text discriminator is very important in meta-training. We also find that removing the style discriminator and the style prototypes result in the performance drop on unseen speaker adaptation. From this result, we find that the style discriminator is important in helping the model adapt to speech from unseen speakers and the style prototypes further enhance the adaptation performance of the model.

6. Conclusion

We have proposed StyleSpeech, a multi-speaker adaptive TTS model which can generate high quality and expressive speech from a single short-duration audio sample of the target speaker. In particular, we propose a style-adaptive layer normalization (SALN) to generate various styles of speech of multiple speakers. Furthermore, we extend StyleSpeech to Meta-StyleSpeech with additional discriminators and meta-learning, to improve its adaptation performance on unseen speakers. Specifically, we simulate one-shot episodic training and the style discriminator utilizes a set of style prototypes to enforce the generator to generate speech that follows the common style of each speaker. The experimental results demonstrate that StyleSpeech and Meta-StyleSpeech can synthesize high-quality speech given the reference audios from both seen and unseen speakers. Moreover, Meta-StyleSpeech achieves significantly improved adaptation performance on the speech from unseen speakers, even with a single reference speech with the length of less than one second. For future work, we plan to further improve Meta-StyleSpeech to perform controllable speech generation by disentangling its latent space, to enhance its practicality in diverse real-world applications.

Acknowledgements This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2019-0-00075, Artificial Intelligence Graduate School Program (KAIST)), and the Engineering Research Center Program through the National Research Foundation of Korea (NRF) funded by the Korean Government MSIT (NRF-2018R1A5A1059921). We sincerely thank the anonymous reviewers for their constructive comments which helped us significantly improve our paper during the rebuttal period.

References


Meta-StyleSpeech: Multi-Speaker Adaptive Text-to-Speech Generation


McFee, B., Raffel, C., Liang, D., Ellis, D., McVicar, M., Battenberg, E., and Nieto, O. librosa: Audio and music signal analysis in python. 2015.


