FastSpeech 2: Fast and High-Quality End-to-End Text to Speech

Yi Ren[1]*, **Chenxu Hu**[1]*, **Xu Tan**[2], **Tao Qin**[2], **Sheng Zhao**[3], **Zhou Zhao**[1]†, **Tie-Yan Liu**[2]
[1]Zhejiang University, China, [2]Microsoft Research Asia, [3]Microsoft Azure Speech
{rayeren,chenxuhu}@zju.edu.cn,{xuta,taoqin,sheng.zhao}@microsoft.com,
zhaozhou@zju.edu.cn,tyliu@microsoft.com

Abstract

Advanced text to speech (TTS) models such as FastSpeech [20] can synthesize speech significantly faster than previous autoregressive models with comparable quality. The training of FastSpeech model relies on an autoregressive teacher model for duration prediction (to provide more information as input) and knowledge distillation (to simplify the data distribution in output), which can ease the one-to-many mapping problem (i.e., multiple speech variations correspond to the same text) in TTS. However, FastSpeech has several disadvantages: 1) the teacher-student distillation pipeline is complicated, 2) the duration extracted from the teacher model is not accurate enough, and the target mel-spectrograms distilled from teacher model suffer from information loss due to data simplification, both of which limit the voice quality. In this paper, we propose FastSpeech 2, which addresses the issues in FastSpeech and better solves the one-to-many mapping problem in TTS by 1) directly training the model with ground-truth target instead of the simplified output from teacher, and 2) introducing more variation information of speech (e.g., pitch, energy and more accurate duration) as conditional inputs. Specifically, we extract duration, pitch and energy from speech waveform and directly take them as conditional inputs during training and use predicted values during inference. We further design FastSpeech 2s, which is the first attempt to directly generate speech waveform from text in parallel, enjoying the benefit of full end-to-end training and even faster inference than FastSpeech. Experimental results show that 1) FastSpeech 2 and 2s outperform FastSpeech in voice quality with much simplified training pipeline and reduced training time; 2) FastSpeech 2 and 2s can match the voice quality of autoregressive models while enjoying much faster inference speed. Audio samples are available at `https://speechresearch.github.io/fastspeech2/`.

1 Introduction

Neural network based text to speech (TTS) has made rapid progress in recent years. Previous neural TTS models such as Tacotron [25], Tacotron 2 [21], Deep Voice 3 [17] and Transformer TTS [10] first generate mel-spectrograms autoregressively from text and then synthesize speech from the generated mel-spectrograms using a separately trained vocoder (e.g., WaveNet [22], WaveGlow [19] and Parallel WaveGAN [27]). They usually suffer from slow inference speed and robustness (word skipping and repeating) issues [20]. In recent years, non-autoregressive TTS models [11, 14, 16, 20] are designed to address these issues, which generate mel-spectrograms with extremely fast speed and avoid robustness issues, while achieving comparable voice quality with previous autoregressive models.

*Equal contribution.
†Corresponding author

Preprint. Under review.

Among those non-autoregressive TTS methods, FastSpeech [20] is one of the most successful models. The training of FastSpeech relies on an autoregressive teacher model to provide 1) the duration of each phoneme to train a duration predictor, and 2) the generated mel-spectrograms for knowledge distillation. While these designs in FastSpeech ease the learning of the one-to-many mapping problem[3] in TTS, they also bring several disadvantages: 1) the two-stage teacher-student distillation pipeline is complicated; 2) the duration extracted from the attention map of the teacher model is not accurate enough, and the target mel-spectrograms distilled from the teacher model suffer from information loss[4] due to data simplification, both of which limit the voice quality and prosody.

In this work, we propose FastSpeech 2 to address the issues in FastSpeech and better handle the one-to-many mapping problem in non-autoregressive TTS. First, to simplify the two-stage teacher-student training pipeline and avoid the information loss due to data simplification, we directly train the FastSpeech 2 model with ground-truth target instead of the simplified output from a teacher. Second, to reduce the information gap between the input (text sequence) and target output (mel-spectrograms) when training non-autoregressive TTS model (input does not contain all the information to predict the target), we introduce some variation information of speech including pitch, energy and more accurate duration into FastSpeech: in training, we extract duration, pitch and energy from the target speech waveform and directly take them as conditional inputs; during inference, we use values predicted by the predictors that are jointly trained with the FastSpeech 2 model. To further simplify the speech synthesis pipeline, we introduce FastSpeech 2s, which abandons mel-spectrograms as intermediate output completely and directly generates speech waveform from text during inference, enjoying the benefit of full end-to-end joint optimization in training and low latency in inference.

FastSpeech 2 and 2s have some connections with other works but show distinctive advantages. Compared with parametric speech synthesis systems such as Merlin [26] and Deep Voice [1], Fastspeech 2 and 2s employ the features such as duration and pitch fully end-to-end, and adopt self-attention based feed-forward network to generate mel-spectrograms or waveform in parallel. Compared with previous non-autoregressive acoustic models [7, 11, 28], most of them focus on improving the duration accuracy to reduce the information gap between the input and output, while FastSpeech 2 and 2s take more missing information into account. Compared with other text-to-waveform models such as ClariNet [18] that jointly train autoregressive acoustic model and non-autoregressive vocoder, FastSpeech 2s employs the fully non-autoregressive architecture. Compared with non-autoregressive vocoders [9, 15, 19, 27], FastSpeech 2s is the first attempt to directly generate waveform from phoneme sequence, instead of linguistic features or mel-spectrograms.

We conduct experiments on the LJSpeech dataset to evaluate FastSpeech 2 and 2s. The results show that 1) FastSpeech 2 outperforms FastSpeech in voice quality and enjoys much simpler training pipeline (3x training time reduction) while inherits its advantages of fast, robust and controllable (even more controllable in pitch and energy) speech synthesis; and 2) both FastSpeech 2 and 2s match the voice quality of autoregressive models and enjoy much faster inference speed. We attach audio samples generated by FastSpeech 2 and 2s at `https://speechresearch.github.io/fastspeech2/`.

2 Method

In this section, we first describe the motivation of the design in FastSpeech 2, and then introduce the architecture of FastSpeech 2, which aims to improve FastSpeech to better handle the one-to-many mapping problem, with simpler training pipeline and higher voice quality.

2.1 Motivation

TTS is a typical one-to-many mapping problem [4, 6, 29], since multiple possible speech sequences can correspond to a text sequence due to different variations in speech audio, such as pitch, duration, sound volume and prosody. In autoregressive TTS, the decoder can condition on the text sequence and the previous mel-spectrograms to predict next mel-spectrograms, where the previous mel-spectrograms can provide some variation information and thus alleviate this problem to a certain

[3] One-to-many mapping problem is described in Section 2.1.

[4] The speech generated by the teacher model loses some variation information about pitch, energy, prosody, etc., and is much simpler and less diverse than the original recording in the training data.

degree. While in non-autoregressive TTS, the only input information is text which is not enough to fully predict the variance in speech. In this case, the model is prone to overfit to the variations of the target speech in the training set, resulting in poor generalization ability.

FastSpeech designs two ways to alleviate the one-to-many mapping problem: 1) Reducing data variance by knowledge distillation in the target side, which can ease the one-to-many mapping problem by simplifying the target. 2) Introducing the duration information (extracted from the attention map of the teacher model) to expand the text sequence to match the length of the mel-spectrogram sequence, which can ease the one-to-many mapping problem by providing more input information. Although knowledge distillation and the duration information extracted from teacher model can improve the training of FastSpeech, they also bring about several issues: 1) The two-stage teacher-student training pipeline makes the training process complicated. 2) The target mel-spectrograms distilled from the teacher model have some information loss compared with the ground-truth ones, since the quality of the audio synthesized from the generated mel-spectrograms is usually worse than that from the ground-truth ones, as shown in Table 1. 3) The duration extracted from the attention map of teacher model is not accurate enough, as analyzed in Table 4a.

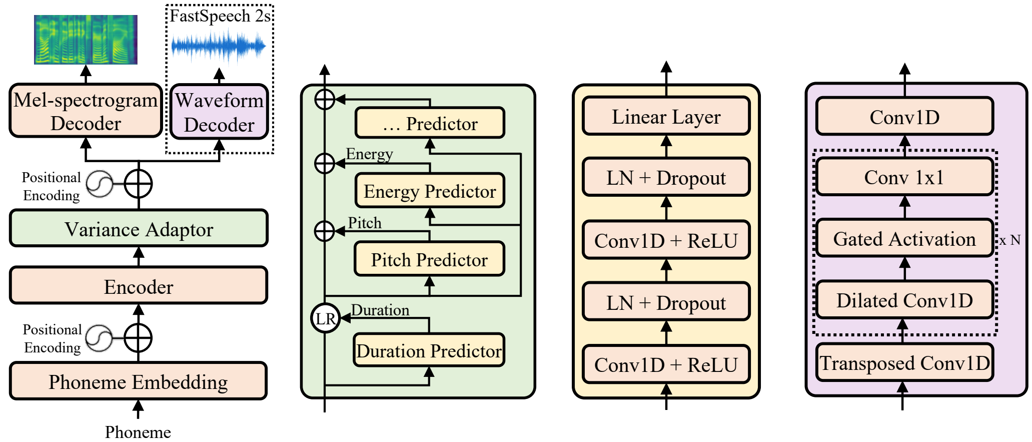
In FastSpeech 2, we address these issues by 1) removing the teacher-student distillation to simplify the training pipeline; 2) using ground-truth speech as the training target to avoid information loss; and 3) improving the duration accuracy and introducing more variance information to ease the one-to-many mapping problem in predicting ground-truth speech. In the following subsection, we introduce the detailed design of FastSpeech 2.

2.2 Model Overview

The overall model architecture of FastSpeech 2 is shown in Figure 1a. The encoder converts the phoneme sequence into the hidden sequence, and then the variance adaptor adds different variance information such as duration, pitch and energy into the hidden sequence, finally the mel-spectrogram decoder converts the adapted hidden sequence into mel-spectrogram sequence in parallel. We use the feed-forward Transformer block, which is a stack of self-attention [24] layer and 1D-convolution as in FastSpeech [20], as the basic structure for the encoder and mel-spectrogram decoder. Different from FastSpeech that relies a teacher-student distillation pipeline and the phoneme duration from a teacher model, FastSpeech 2 makes several improvements. First, we remove the teacher-student distillation pipeline, and directly use ground-truth mel-spectrograms as target for model training, which can avoid the information loss in distilled mel-spectrograms and increase the upper bound of the voice quality. Second, our variance adaptor consists of not only length regulator but also pitch and energy predictors, where 1) the length regulator uses the phoneme duration obtained by forced alignment [13], which is more accurate than that extracted from the attention map of autoregressive teacher model; and 2) the additional pitch and energy predictors can provide more variance information, which is important to ease the one-to-many mapping problem in TTS. Third, to further reduce the training pipeline and push it towards a fully end-to-end system, we propose FastSpeech 2s, which directly generates waveform from text, without cascaded mel-spectrogram generation (acoustic model) and waveform generation (vocoder). In the following subsections, we describe the detailed design of the variance adaptor and direct waveform generation in our method.

2.3 Variance Adaptor

The variance adaptor aims to add variance information (e.g., duration, pitch, energy, etc.) to the phoneme hidden sequence, which can provide enough information to predict variant speech for the one-to-many mapping problem in TTS. As shown in Figure 1b, the variance adaptor consists of 1) duration predictor (i.e., length regulator, as used in FastSpeech), 2) pitch predictor, and 3) energy predictor. More variance information can be added in the variance adaptor, which is discussed in the following paragraph. During training, we take the ground-truth value of duration, pitch and energy extracted from the recordings as input into the hidden sequence to predict the target speech. At the same time, we use separate variance predictors for duration, pitch and energy predictions, which are used during inference to synthesize target speech. In this subsection, we first describe the model details of variance predictor, and then describe how to leverage the duration, pitch and energy information in the variance adaptor.



(a) FastSpeech 2 (b) Variance adaptor (c) Variance predictor (d) Waveform decoder

Figure 1: The overall architecture for FastSpeech 2 and 2s. LR in subfigure (b) denotes the length regulator operation proposed in FastSpeech. LN in subfigure (c) denotes layer normalization. Variance predictor represents duration/pitch/energy predictor.

Variance Predictor As shown in Figure 1c, variance predictor has the similar model structure as the duration predictor in FastSpeech, which takes the hidden sequence as input and predicts the variance of each phoneme (duration) or frame (pitch and energy) with the mean square error (MSE) loss. Variance predictor consists of a 2-layer 1D-convolutional network with ReLU activation, each followed by the layer normalization and the dropout layer, and an extra linear layer to project the hidden states into the output sequence. For the duration predictor, the output is the length of each phoneme in the logarithmic domain. For the pitch predictor, the output sequence is the frame-level fundamental frequency sequence (F_0). For the energy predictor, the output is a sequence of the energy of each mel-spectrogram frame. All predictors share the same model structure but not model parameters.

Details of Variance Information In addition to text, speech audio usually contains a lot of other variance information including 1) phoneme duration, which represents how fast the speech voice sounds; 2) pitch, which is a key feature to convey emotions and greatly affects the perception; 3) energy, which indicates frame-level magnitude of mel-spectrograms and directly affects the loss computed on mel-spectrograms; 4) emotion, style, speaker and so on. The variance information is not determined entirely by the text and thus harms the training of non-autoregressive TTS model due to the one-to-many mapping problem. In this paragraph, we describe the details of how we use pitch, energy and duration in the variance adaptor.

- **Duration** To improve the alignment accuracy and thus reduce the information gap between the model input and output, instead of extracting the phoneme duration using a pre-trained autoregressive TTS model in FastSpeech, we extract the phoneme duration with MFA [13], an open-source system for speech-text alignment with good performance, which can be trained on paired text-audio corpus without any manual alignment annotations. We convert the alignment results generated by MFA to the phoneme-level duration sequence and feed it into the length regulator to expand the hidden states of the phoneme sequence.
- **Pitch and Energy** We extract F_0 from the raw waveform[5] with the same hop size of target mel-spectrograms to obtain the pitch of each frame, and compute L2-norm of the amplitude of each STFT frame as the energy. Then we quantize F_0 and energy of each frame to 256 possible values[6] and encoded them into a sequence of one-hot vectors (p and e) respectively. In the training process, we lookup the pitch and energy embedding with p and e and add them to the hidden sequence. The pitch and energy predictors directly predict the values of F_0 and energy instead of the one-hot vector and are optimized with mean square error. During inference, we predict the F_0 and energy using variance predictors.

[5]We extract the $F0$ using PyWorldVocoder from `https://github.com/JeremyCCHsu/Python-Wrapper-for-World-Vocoder`. We do not introduce voiced/unvoiced flags and directly set F_0 to 0 for unvoiced frames for simplicity. We find it does not affect the quality of synthesized speech.

[6]We use log-scale bins for F_0 and uniform bins for energy.

2.4 FastSpeech 2s

To simplify the text-to-waveform generation pipeline and enable fully end-to-end training and inference in text-to-waveform generation, in this subsection, we propose FastSpeech 2s, which directly generates waveform from text, without cascaded mel-spectrogram generation (acoustic model) and waveform generation (vocoder). We first discuss the challenges lie in non-autoregressive text-to-waveform generation, then describe details in FastSpeech 2s, including model structure and training and inference process.

Challenges in Text-to-Waveform Generation When pushing TTS pipeline towards fully end-to-end framework, there are several challenges: 1) Since waveform contains more variance information (e.g., phase) than mel-spectrograms, the information gap between the input and output is larger than that in text-to-spectrogram generation. 2) It is difficult to train on the audio clip that corresponds to the full text sequence due to the extremely long waveform samples and limited memory. As a result, we can only train on a short audio clip that corresponds to a partial text sequence which makes it hard for the model to capture the relationship among phonemes in different partial text sequences and thus harms the text feature extraction.

Our Method To tackle the challenges above, we make several designs in the waveform decoder: 1) Considering the phase information is difficult to predict using a variance predictor [3], we introduce adversarial training in the waveform decoder to force it to implicitly recover the phase information by itself [27]. 2) We leverage the mel-spectrogram decoder which is trained on the full text sequence to help on the text feature extraction. As shown in Figure 1d, the waveform decoder is based on the stucture of WaveNet [22] including non-causal convolutions and gated activation [23]. The waveform decoder takes a sliced hidden sequence corresponding to a short audio clip as input and upsamples it with transposed 1D-convolution to match the length of audio clip. The discriminator in the adversarial training adopts the same structure in Parallel WaveGAN [27]. The waveform decoder is optimized by the multi-resolution STFT loss computed by the sum of several different STFT losses and the discriminator loss following Parallel WaveGAN. During inference, we discard the mel-spectrogram decoder and only use the waveform decoder to synthesize speech audio.

3 Experiments and Results

3.1 Experimental Setup

Datasets We evaluate FastSpeech 2 on LJSpeech dataset [5]. LJSpeech contains 13,100 English audio clips (about 24 hours) and corresponding text transcripts. We split the dataset into three sets: 12,228 samples for training, 349 samples (with document title LJ003) for validation and 523 samples (with document title LJ001 and LJ002) for testing. To alleviate the mispronunciation problem, we convert the text sequence into the phoneme sequence [1, 21, 25] with an open-source grapheme-to-phoneme tool[7]. We transform the raw waveform into mel-spectrograms following Shen et al. [21] and set frame size and hop size to 1024 and 256 with respect to the sample rate 22050.

Model Configuration Our FastSpeech 2 consists of 4 feed-forward Transformer (FFT) blocks [20] in the encoder and the mel-spectrogram decoder. In each FFT block, the dimension of phoneme embeddings and the hidden size of the self-attention are set to 256. The number of attention heads is set to 2 and the kernel sizes of the 1D-convolution in the 2-layer convolutional network after the self-attention layer are set to 9 and 1, with input/output size of 256/1024 for the first layer and 1024/256 in the second layer. The output linear layer converts the 256-dimensional hidden states into 80-dimensional mel-spectrograms and optimized with mean absolute error (MAE). The size of the phoneme vocabulary is 76, including punctuations. In the variance predictor, the kernel sizes of the 1D-convolution are set to 3, with input/output sizes of 256/256 for both layers and the dropout rate is set to 0.5. Our waveform decoder consists of 1-layer transposed 1D-convolution with filter size 64 and 30 layers of dilated residual convolution blocks, whose skip channel size and kernel size of 1D-convolution are set to 64 and 3. The configurations of the discriminator in FastSpeech 2s are the same as Parallel WaveGAN [27]. We list hyperparameters and configurations of all models used in our experiments in Appendix A.

[7] `https://github.com/Kyubyong/g2p`

Training and Inference We train FastSpeech 2 on 1 NVIDIA V100 GPU, with batchsize of 48 sentences. We use the Adam optimizer [8] with $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\varepsilon = 10^{-9}$ and follow the same learning rate schedule in [24]. It takes 160k steps for training until convergence. In the inference process, the output mel-spectrograms of our FastSpeech 2 are transformed into audio samples using pre-trained Parallel WaveGAN [27][8]. For FastSpeech 2s, we train the model on 2 NVIDIA V100 GPUs, with batchsize of 6 sentences on each GPU. The waveform decoder takes the sliced hidden states corresponding to 20,480 waveform sample clips as input. The optimizer and learning rate schedule for FastSpeech 2s are the same as FastSpeech 2. The details of the adversarial training follow Parallel WaveGAN [27]. It takes 600k steps for training until convergence for FastSpeech 2s.

3.2 Results

In this section, we first evaluate the audio quality, training and inference speedup of FastSpeech 2 and 2s. Then we conduct analyses and ablation studies of our method.

3.2.1 Performance of FastSpeech 2

Method	MOS
GT	4.27 ± 0.07
GT (Mel + PWG)	3.92 ± 0.08
Tacotron 2 [21] (Mel + PWG)	3.74 ± 0.07
Transformer TTS [10] (Mel + PWG)	**3.79 ± 0.08**
FastSpeech [20] (Mel + PWG)	3.67 ± 0.08
FastSpeech 2 (Mel + PWG)	3.77 ± 0.08
FastSpeech 2s	**3.79 ± 0.08**

Table 1: The MOS with 95% confidence intervals.

Audio Quality To evaluate the perceptual quality, we perform mean opinion score (MOS) [2] evaluation on the test set. Twenty native English speakers are asked to make quality judgments about the synthesized speech samples. The text content keeps consistent among different systems so that all testers only examine the audio quality without other interference factors. We compare the MOS of the audio samples generated by *FastSpeech 2* and *FastSpeech 2s* with other systems, including 1) *GT*, the ground-truth recordings; 2) *GT (Mel + PWG)*, where we first convert the ground-truth audio into mel-spectrograms, and then convert the mel-spectrograms back to audio using Parallel WaveGAN [27] (PWG); 3) *Tacotron 2 [21] (Mel + PWG)*; 4) *Transformer TTS [10] (Mel + PWG)*. 5) *FastSpeech [20] (Mel + PWG)*. All the systems in 3), 4) and 5) use Parallel WaveGAN as the vocoder for fair comparison. The results are shown in Table 1. It can be seen that our FastSpeech 2 and 2s can match the voice quality of autoregressive models *Transformer TTS* and *Tacotron 2*. Importantly, FastSpeech 2 and 2s outperform FastSpeech, which demonstrates the effectiveness of providing variance information such as pitch, energy and more accurate duration and directly taking ground-truth speech as training target without using teacher-student distillation pipeline.

Method	Training Time (h)	Inference Speed (RTF)	Inference Speedup
Transformer TTS [10]	38.64	8.26×10^{-1}	/
FastSpeech [20]	53.12	5.41×10^{-3}	152×
FastSpeech 2	**16.46**	5.51×10^{-3}	149×
FastSpeech 2s	/	$\mathbf{4.87 \times 10^{-3}}$	**170×**

Table 2: The comparison of training time and inference latency in waveform synthesis. The training time of *FastSpeech* includes teacher and student training. RTF denotes the real-time factor, that is the time (in seconds) required for the system to synthesize one second waveform. The training and inference latency test is conducted on a server with 36 Intel Xeon CPU, 256GB memory, 1 NVIDIA V100 GPU and batch size of 48 for training and 1 for inference. Besides, we do not include the time of GPU memory garbage collection and transferring input and output data between the CPU and the GPU. The speedup in waveform synthesis for FastSpeech is larger than that reported in Ren et al. [20] since we use Parallel WaveGAN as the vocoder which is much faster than WaveGlow.

Training and Inference Speedup FastSpeech 2 simplifies the training pipeline of FastSpeech by removing the teacher-student distillation process, and thus reduces the training time. We list the total training time of *Transformer TTS* (the autoregressive teacher model), FastSpeech (including the training of *Transformer TTS* teacher model and *FastSpeech* student model) and *FastSpeech 2* in Table 2. It can be seen that FastSpeech 2 reduces the total training time by 3.22× compared with

[8] `https://github.com/kan-bayashi/ParallelWaveGAN`

FastSpeech. Note that training time here only includes acoustic model training, without considering the vocoder training. Therefore, we do not compare the training time of FastSpeech 2s here. We then evaluate the inference latency of FastSpeech 2 and 2s compared with the autoregressive Transformer TTS model, which has the similar number of model parameters with FastSpeech 2 and 2s. We show the inference speedup for waveform generation in Table 2. It can be seen that compared with the Transformer TTS model, FastSpeech 2 and 2s speeds up the audio generation by 149× and 170× respectively in waveform synthesis, which shows that FastSpeech 2s is faster than FastSpeech and FastSpeech 2 due to full end-to-end generation and the removal of the mel-spectrogram decoder.

3.2.2 Analyses on Variance Information

Better Optimization and Generalization To analyze the impact of introducing variance information on the optimization and generalization of the model, we plot the mel-spectrogram loss curves of *FastSpeech* and *FastSpeech 2* on the training and validation set in Figure 2. From the training loss curves, we can see that the training loss of *FastSpeech 2* is smaller than *FastSpeech*, demonstrating that the provided variance information (pitch, energy and more accurate duration) can help model optimization. From the gap between training and validation loss curves of each model, we can see that the training and validation loss gap of FastSpeech 2 (≈ 0.037 at 160k steps) is smaller than FastSpeech (≈ 0.119 at 160k steps), which indicates that introducing variance information (pitch, energy and more accurate duration) can improve generalization.

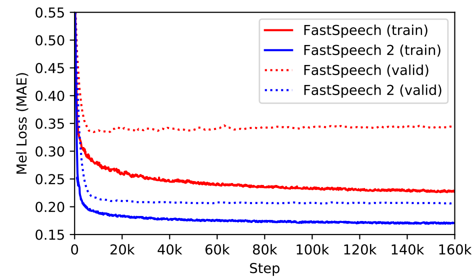


Figure 2: The training and validation loss curves of FastSpeech and FastSpeech 2.

More Accurate Variance Information in Synthesized Speech To verify whether providing more variance information (e.g., pitch and energy) as input can indeed synthesize speech with more accurate pitch and energy, we compare the accuracy of pitch and energy of the synthesized speech by FastSpeech and FastSpeech 2. We compute the accuracy by calculating the mean absolute error (MAE) between the frame-wise pitch/energy extracted from the generated waveform and the ground-truth speech. To ensure the numbers of frames in the synthesized and ground-truth speech are the same, we use the ground-truth duration extracted by MFA in both FastSpeech and FastSpeech 2. The results are shown in Table 3. It can be seen that compared with FastSpeech, FastSpeech 2 and 2s can both synthesize speech audio with more similar pitch and energy to the ground-truth audio.

Method	Pitch	Energy
FastSpeech [20]	21.67	0.142
FastSpeech 2	20.30	**0.131**
FastSpeech 2s	**20.28**	0.133

Table 3: The mean absolute error of the pitch and energy in synthesized speech audio.

More Accurate Duration for Model Training We then analyze the accuracy of the provided duration information to train the duration predictor and the effectiveness of more accurate duration for better voice quality. We manually align 50 audio and the corresponding text in phoneme level and get the ground-truth phoneme-level duration. We compute the average of absolute phoneme boundary differences [13] using the duration from the teacher model of FastSpeech and from MFA we used in this paper respectively. The results are shown in Table 4a. We can see that MFA can generate more accurate duration than the teacher model of FastSpeech. Next, we replace the duration used in FastSpeech (from teacher model) with that extracted by MFA and conduct the CMOS [12] test to compare the voice quality between two FastSpeech models trained with different durations. The results are listed in Table 4b and it can be seen that more accurate duration information improves the voice quality of FastSpeech, which verifies the effectiveness of our improved duration from MFA.

Method	Δ (ms)
Duration from teacher model	19.68
Duration from MFA	12.47

(a) Alignment accuracy comparison.

Setting	CMOS
FastSpeech + Duration from teacher	0
FastSpeech + Duration from MFA	+0.195

(b) CMOS comparison.

Table 4: The comparison of the duration from teacher model and MFA. Δ means the average of absolute boundary differences.

3.2.3 Ablation Study

In this subsection, we conduct ablation studies to demonstrate the effectiveness of several variance information of FastSpeech 2 and 2s, including pitch and energy[9]. We conduct CMOS evaluation for these ablation studies. The results are shown in Table 5. We find that removing the energy variance (Row 2 in both subtables) in FastSpeech 2 and 2s results in performance drop in terms of voice quality (-0.045 and -0.150 CMOS respectively), indicating that energy variance can slightly improve the voice quality for FastSpeech 2, but more effective for FastSpeech 2s. We also find that removing the pitch variance (Row 3 in both subtables) in FastSpeech 2 and 2s results in -0.230 and -1.045 CMOS respectively, which demonstrates the effectiveness of pitch variance. When we remove both pitch and energy variance (Row 4 in both subtables), the voice quality further drops, indicating that pitch and energy variance together help improve the performance of FastSpeech 2 and 2s.

Setting	CMOS
FastSpeech 2	0
FastSpeech 2 - energy	-0.045
FastSpeech 2 - pitch	-0.230
FastSpeech 2 - pitch - energy	-0.385

(a) CMOS comparison for FastSpeech 2.

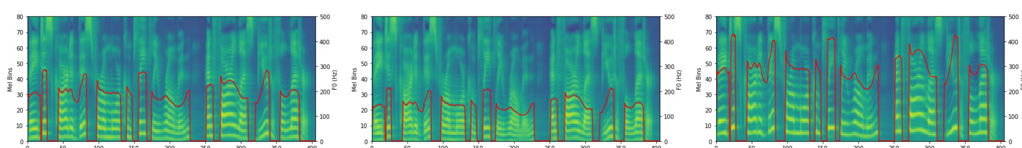
Setting	CMOS
FastSpeech 2s	0
FastSpeech 2s - energy	-0.150
FastSpeech 2s - pitch	-1.045
FastSpeech 2s - pitch - energy	-1.070

(b) CMOS comparison for FastSpeech 2s.

Table 5: CMOS comparison in the ablation studies.

3.2.4 Variance Control

FastSpeech 2 and 2s introduce several variance information to ease the one-to-many mapping problem in TTS. As a byproduct, they also make the synthesized speech more controllable. As a demonstration, we manipulate pitch input to control the pitch in synthesized speech in this subsubsection. We show the mel-spectrograms before and after the pitch manipulation in Figure 3. From the samples, we can see that FastSpeech 2 generates high-quality mel-spectrograms after adjusting the $\hat{F}_0$.



(a) $\hat{F}_0 = F_0$ (b) $\hat{F}_0 = 0.75F_0$ (c) $\hat{F}_0 = 1.50F_0$

Figure 3: The mel-spectrograms of the voice with different $\hat{F}_0$. F_0 is the fundamental frequency of original audio. The red curves denote $\hat{F}_0$ contours. The input text is "*They discarded this for a more completely Roman and far less beautiful letter.*"

4 Conclusion

In this work, we proposed FastSpeech 2, a fast and high-quality end-to-end TTS system, to address the issues in FastSpeech and ease the one-to-many mapping problem: 1) we directly train the model with ground-truth mel-spectrograms to simplify the training pipeline and also avoid information loss compared with FastSpeech; and 2) we improve the duration accuracy and introduce more variance information including pitch and energy to ease the one-to-many mapping problem. Moreover, based on FastSpeech 2, we further developed FastSpeech 2s, a non-autoregressive text-to-waveform generation model, which enjoys the benefit of full end-to-end training and inference. Our experimental results show that FastSpeech 2 and 2s can surpass FastSpeech in terms of voice quality, with much simpler training pipeline while inheriting the advantages of fast, robust and controllable speech synthesis of FastSpeech. In the future, we will consider more variance information to further improve the voice quality and will further speed up the inference with more light-weight model.

[9]We do not study duration information since duration is a necessary for FastSpeech and FastSpeech 2. Besides, we have already analyzed the effectiveness of our improved duration in the last paragraph.

References

[1] Sercan O Arik, Mike Chrzanowski, Adam Coates, Gregory Diamos, Andrew Gibiansky, Yongguo Kang, Xian Li, John Miller, Andrew Ng, Jonathan Raiman, et al. Deep voice: Real-time neural text-to-speech. *arXiv preprint arXiv:1702.07825*, 2017.

[2] Min Chu and Hu Peng. Objective measure for estimating mean opinion score of synthesized speech, April 4 2006. US Patent 7,024,362.

[3] Jesse Engel, Lamtharn Hantrakul, Chenjie Gu, and Adam Roberts. Ddsp: Differentiable digital signal processing. *arXiv preprint arXiv:2001.04643*, 2020.

[4] Michael Gadermayr, Maximilian Tschuchnig, Dorit Merhof, Nils Krämer, Daniel Truhn, and Burkhard Gess. An asymetric cycle-consistency loss for dealing with many-to-one mappings in image translation: A study on thigh mr scans. *arXiv preprint arXiv:2004.11001*, 2020.

[5] Keith Ito. The lj speech dataset. `https://keithito.com/LJ-Speech-Dataset/`, 2017.

[6] Chrisina Jayne, Andreas Lanitis, and Chris Christodoulou. One-to-many neural network mapping techniques for face image synthesis. *Expert Systems with Applications*, 39(10): 9778–9787, 2012.

[7] Jaehyeon Kim, Sungwon Kim, Jungil Kong, and Sungroh Yoon. Glow-tts: A generative flow for text-to-speech via monotonic alignment search. *arXiv preprint arXiv:2005.11129*, 2020.

[8] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[9] 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 *Advances in Neural Information Processing Systems*, pages 14881–14892, 2019.

[10] Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, and Ming Liu. Neural speech synthesis with transformer network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6706–6713, 2019.

[11] Dan Lim, Won Jang, Hyeyeong Park, Bongwan Kim, Jesam Yoon, et al. Jdi-t: Jointly trained duration informed transformer for text-to-speech without explicit alignment. *arXiv preprint arXiv:2005.07799*, 2020.

[12] Philipos C Loizou. Speech quality assessment. In *Multimedia analysis, processing and communications*, pages 623–654. Springer, 2011.

[13] Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger. Montreal forced aligner: Trainable text-speech alignment using kaldi. In *Interspeech*, pages 498–502, 2017.

[14] Chenfeng Miao, Shuang Liang, Minchuan Chen, Jun Ma, Shaojun Wang, and Jing Xiao. Flow-tts: A non-autoregressive network for text to speech based on flow. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7209–7213. IEEE, 2020.

[15] Aaron van den Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George van den Driessche, Edward Lockhart, Luis C Cobo, Florian Stimberg, et al. Parallel wavenet: Fast high-fidelity speech synthesis. *arXiv preprint arXiv:1711.10433*, 2017.

[16] Kainan Peng, Wei Ping, Zhao Song, and Kexin Zhao. Parallel neural text-to-speech. *arXiv preprint arXiv:1905.08459*, 2019.

[17] Wei Ping, Kainan Peng, Andrew Gibiansky, Sercan O. Arik, Ajay Kannan, Sharan Narang, Jonathan Raiman, and John Miller. Deep voice 3: 2000-speaker neural text-to-speech. In *International Conference on Learning Representations*, 2018.

[18] Wei Ping, Kainan Peng, and Jitong Chen. Clarinet: Parallel wave generation in end-to-end text-to-speech. In *International Conference on Learning Representations*, 2019.

[19] Ryan Prenger, Rafael Valle, and Bryan Catanzaro. Waveglow: A flow-based generative network for speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3617–3621. IEEE, 2019.

[20] 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 *Advances in Neural Information Processing Systems*, pages 3165–3174, 2019.

[21] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4779–4783. IEEE, 2018.

[22] Aäron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *SSW*, 125, 2016.

[23] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in neural information processing systems*, pages 4790–4798, 2016.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[25] Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al. Tacotron: Towards end-to-end speech synthesis. *arXiv preprint arXiv:1703.10135*, 2017.

[26] Zhizheng Wu, Oliver Watts, and Simon King. Merlin: An open source neural network speech synthesis system. In *SSW*, pages 202–207, 2016.

[27] Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim. Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6199–6203. IEEE, 2020.

[28] Zhen Zeng, Jianzong Wang, Ning Cheng, Tian Xia, and Jing Xiao. Aligntts: Efficient feed-forward text-to-speech system without explicit alignment. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6714–6718. IEEE, 2020.

[29] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *Advances in neural information processing systems*, pages 465–476, 2017.

Appendix A Model Hyperparameters

Table 6: Hyperparameters of Transformer TTS, FastSpeech and FastSpeech 2. Decoder is for Transformer TTS, mel-spectrogram decoder is for FastSpeech and FastSpeech 2 and waveform decoder is for FastSpeech 2s.

Hyperparameter	**Transformer TTS**	**FastSpeech/FastSpeech 2/2s**
Phoneme Embedding Dimension	256	256
Pre-net Layers	3	/
Pre-net Hidden	256	/
Encoder Layers	4	4
Encoder Hidden	256	256
Encoder Conv1D Kernel	9	9
Encoder Conv1D Filter Size	1024	1024
Encoder Attention Heads	2	2
Decoder/Mel-Spectrogram Decoder Layers	4	4
Decoder/Mel-Spectrogram Decoder Hidden	384	384
Decoder/Mel-Spectrogram Decoder Conv1D Kernel	9	9
Decoder/Mel-Spectrogram Decoder Conv1D Filter Size	1024	1024
Decoder/Mel-Spectrogram Decoder Attention Headers	2	2
Encoder/Decoder Dropout	0.1	0.2
Variance Predictor Conv1D Kernel	/	3
Variance Predictor Conv1D Filter Size	/	256
Variance Predictor Dropout	/	0.5
Waveform Decoder Convolution Blocks	/	30
Waveform Decoder Dilated Conv1D Kernel size	/	3
Waveform Decoder Transposed Conv1D Filter Size	/	64
Waveform Decoder Skip Channlel Size	/	64
Batch Size	48	48/48/12
Total Number of Parameters	24M	23M/27M/28M