IMPROVING FASTSPEECH TTS WITH EFFICIENT SELF-ATTENTION AND COMPACT FEED-FORWARD NETWORK

Yujia Xiao, Xi Wang, Lei He, Frank K. Soong

Microsoft China

ABSTRACT

FastSpeech, as a feed-forward transformer based TTS, can avoid the slow serial, autoregressive inference to generate the target mel-spectrogram in a parallel way. As a non-autoregressive TTS, the latency and computation load in inference is shifted from vocoder to transformer where the efficiency is limited by the quadratic time and memory complexity in the self-attention mechanism, particularly for a long text sequence. To tackle this challenges, We propose two models, ProbSparseFS and LinearizedFS, which have efficient self-attention arrangements to improve the inference speed and memory complexity. LinearizedFS has achieved 3.4x memory savings and 2.1x inference speedup, compared with the those of the baseline FastSpeech. A further optimized LinearizedFS with a lightweight FFN can accelerate the inference speed by 3.6x more. We do subjective voice quality evaluations in MOS and CMOS of News report and Audiobook applications, for multi-speaker and multi-style scenarios. Test results verified that the proposed models yield a TTS quality which is on-par with that of the baseline system but with much better memory efficiency and inference speed.

Index Terms— Speech Synthesis, Efficient Transformer, Self-Attention

1. INTRODUCTION

With the rapid development of neural speech synthesis, end-to-end acoustic models like Tacotron 1/2 [18, 14], DeepVoice 1/2 [1, 3], Transformer TTS [8] together with vocoder models like WaveNet [10], WaveRNN [4] can generate realistic speech close to recording. However, the inference speed of these models are limited by their autoregressive structure. Non-autogressive vocoder model like Parallel WaveNet [9], WaveGlow [11], and MelGAN [7] largely relieved the sequential inference in samples. On the other hand, non-autoregressive acoustic model like FastSpeech [13], which is a Transformer [16] based feed-forward network, can do frame level inference in parallel. Without the auto-regressive structure limitation, the bottleneck of inference efficiency is shifted from vocoder to acoustic model. For example, in a typic TTS system like FastSpeech with MelGAN, the Transformer based FastSpeech occupies around 2/3 computational time cost. Based on that observation, we hope to improve the efficiency of acoustic model in further.

The global receptive field of self-attention based Transformer block benefits both FastSpeech encoder and decoder for better speech synthesis quality. However, standard self-attention mechanism shows poor scalability, especially for long sequences, due to the quadratic time and memory complexity. Therefore, in TTS scenario, text to be synthesized usually needs to be separated into sentences or even phrases if the length exceeds a threshold. To alleviate this limitation, we need a more efficient transformer block while maintaining the same TTS voice quality.

To address the efficiency issue of self-attention, various model variants have been proposed to improve the model memory and computational cost. A comprehensive survey of the recent efficient Transformer models is reviewed in [15], which presents efficient Transformers in different categories. For example, Sparse Transformer [2] is the typical one to leverage the fixed local and stride attention patterns. It yields a limited speed-up but at a cost of steep performance degradation [12]. Reformer [6] is based on locality sensitive hashing and introduces reversible Transformer layers to save memory cost further. This method is effective only for extremely long sequences. In Linformer [17], the stochastic matrix formed by self-attention is approximated by a low-rank matrix. However, the complexity may revert to $O(N^2)$ since the projection matrix cannot be fixed for input.

This paper investigates two efficient self-attention blocks based on the FastSpeech model. One is ProbSprase Attention from Informer [19], which is proposed to deal with long sequences, can reduce the time complexity and memory usage to $O(NlnN)$. The other one is Linearized Attention from Linear Transformer [5], which is based on a kernel-based formulation and compute in linear time with a constant memory cost. In addition to optimization on self-attention module, we also simplify the Feed-Forward Network (FFN) block to further improve the efficiency. We applied the proposed models to both English and Chinese TTS in multi-style, expressive voices. Evaluation on both quality (MOS and CMOS) and performance (Stress Test on Memory; Performance Test on Time) validates the proposed models can synthesis long sequences efficiently with almost no degradation of voice quality, when compared with the baseline system.

978-1-6654-0540-9/22/$31.00 ©2022 IEEE

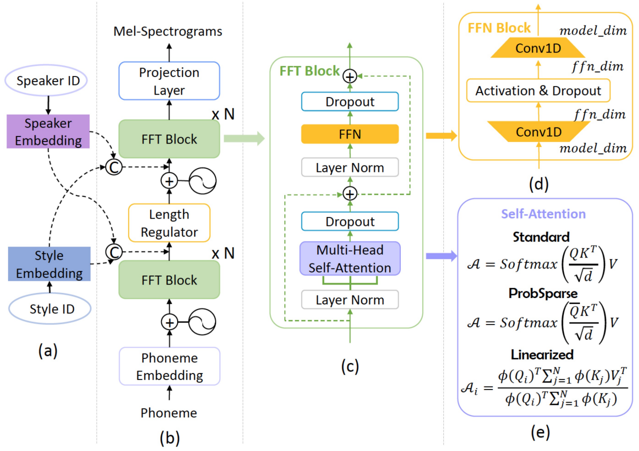

Fig. 1. a) Multi-Speaker / Multi-Style Component b) Fastspeech Framework c) FFT block d) FFN Block e) Standard and Efficient self-attention modules.

2. FASTSPEECH

FastSpeech is a non-autoregressive mel-spectrogram generation model which uses the same encoder and decoder as illustrated in Fig 1-(b). With using a duration predictor and a length regulator, FastSpeech avoids the slow autoregressive inference and can generate mel-spectrogram parallelly. Specifically, the encoder or decoder is formed by cascading several stacked Feed Forward Transformer (FFT) blocks. Each FFT block is made of a multi-head self-attention and a FFN block as shown in Fig 1-(c). In Fig 1-(d), the FFN block is configured with a 2-layer 1D convolutional network with ReLU activation which can capture the text and mel-spectrogram information from adjacent hidden states. Fig 1-(e) presents the standard self-attention and its two efficient variants, ProbSparse and Linearized self-attentive mechanism, which are elaborated in Session 3.

2.1. FastSpeech for Multi-Speakers and Multi-Styles

Multi-speaker neural TTS is practical due to its robustness and efficient deployment. In such paradigm, we leverage the extra data from multi-speakers to build a robust model which can synthesize a target voice with only limited training data. Using a unified model for synthesizing multi-speaker voices is obviously more efficient than modeling individual speaker separately. The same concept of multi-speaker TTS model can be similarly extended to multi-style one to achieve scalability. Our proposed model is evaluated for both multi-speaker and multi-style scenario. Both speaker and style information is encoded with latent representations as shown in Fig 1-(a), in which two correpsonding embeddings are concatenated with the encoder's output and decoder's input.

3. EFFICIENT TRANSFORMER BLOCK

The standard self-attention received linear transformations of input sequences, a tuple input (query, key and value), and then learn an attention matrix by a Softmax function as shown in Eq (1). In this formulation, $Q, K, V \in \mathcal{R}^{N \times d}$, where N is the input sequence length and d is the dimensionality after the projection. As the dot product between each element in the Q and K is taken to learn from each other, the memory and computational complexity is $O(N^2)$. For speech synthesis, this will limit the inference capacity due to the frame-based speech feature, like mel-spectrogram. We will discuss two efficient self-attention variants in the following sub-sections.

$$\mathcal{A}(Q, K, V) = Softmax(QK^T/\sqrt{d})V \tag{1}$$

3.1. ProbSparse Self-Attention

Based on the assumption that the canonical self-attention is sparse, ProbSparse method aims to improve the efficiency by ignoring the workload of unimportant dot-product pairs, i.e., figuring out the important ones. The i-th query's attention over the j-th key is described as Eq (2).

$$p(K_j|Q_i) = \frac{exp(Q_i K_j^T/\sqrt{d})}{\sum_{n=1}^{N} exp(Q_i K_n^T/\sqrt{d})} \tag{2}$$

By using Kullback-Leibler divergence to measure the correlation between two distributions, the i-th query's sparsity measurement is defined in Eq (3), where a larger $M(Q_i, K)$ means a higher probability of dominating dot-product.

$$M(Q_i, K) = ln \sum_{j=1}^{N} exp(\frac{Q_i K_j^T}{\sqrt{d}}) - \frac{1}{N} \sum_{j=1}^{N} \frac{Q_i K_j^T}{\sqrt{d}} \tag{3}$$

However, to compare different query's sparsity, we need to calculate all dot-product pairs according to Eq (3). To avoid the traversal calculation and the potential numerical stability issue caused by Log-Sum-Exp in the first item, a max-mean approximation measurement is proposed as Eq (4).

$$\bar{M}(Q_i, K) = max_j\{\frac{Q_i K_j^T}{\sqrt{d}}\} - \frac{1}{N} \sum_{j=1}^{N} \frac{Q_i K_j^T}{\sqrt{d}} \tag{4}$$

Based on Eq (4) and proof details in [19], only $U = NlnN$ random selected dot-product pairs are needed to calculate the $\bar{M}(Q_i, K)$, which optimizes the computational and memory cost with complexity as $O(NlnN)$. Then, sorting out the Top-u queries, $\bar{Q}$, to attend the attention matrix calculation. Here $u = c \cdot lnN$ where c is a constant sampling factor.

3.2. Linearized Self-Attention

The self-attention function can be regarded as similarity evaluation between each item in the input sequence. For the i-th query, a generalized attention equation for any similarity function can be represented as Eq (5). The standard self-attention, Eq (1), can be described by setting the similarity function $sim(Q_i, K_j) = exp(Q_i K_j^T/\sqrt{d})$.

$$\mathcal{A}(Q_i, K, V) = \frac{\sum_{j=1}^{N} sim(Q_i, K_j)V_j}{\sum_{j=1}^{N} sim(Q_i, K_j)} \tag{5}$$

The similarity function can be replaced by other attention functions, the only contraint is that the function should be non-negative. Given a qualified kernel with a row-wise feature map $\phi(x)$, Eq (5) can be rewrote as Eq (6).

$$\mathcal{A}(Q_i, K, V) = \frac{\sum_{j=1}^{N} \phi(Q_i)^T \phi(K_j) V_j}{\sum_{j=1}^{N} \phi(Q_i)^T \phi(K_j)} \tag{6}$$

By using the associative property of matrix multiplication, the attention equation can be simplified as Eq (7), where the calculation of $\sum_{j=1}^{N} \phi(K_j)V_j$ and $\sum_{j=1}^{N} \phi(K_j)$ is independent from queries. Therefore, these two terms can be computed beforehand and reused for each query, which reduces the computational and memory complexity to $O(N)$.

$$\mathcal{A}(Q_i, K, V) = \frac{\phi(Q_i)^T \sum_{j=1}^{N} \phi(K_j) V_j^T}{\phi(Q_i)^T \sum_{j=1}^{N} \phi(K_j)} \tag{7}$$

To ensure a positive similarity and avoid 0 gradients when x is negative, the elu function is chosen and the feature map function is defined as $\phi(x) = elu(x) + 1$.

3.3. Lightweight FFN

Besides the self-attention module, the FFN block described in Fig 1-(d) also consumes considerable time and memory in the whole structure, since the inner-layer dimension (ffn_{dim}) is often much larger than the input and output dimension ($model_{dim}$). We assume that after replacing the canonical self-attention with more efficient ones, the information gathered by attention matrix is purified. Therefore, it is feasible to reduce the dimensionality of ffn_{dim} while keeping the same learning capability. Such a lightweight FFN further improves the model efficiency.

4. EXPERIMENTS

In this session, we conduct extensive experiments to evaluate our models from both quality and performance aspects. For voice quality evaluation, we use CMOS (Comparison Mean Opinion Score) and MOS (Mean Opinion Score) subjective tests. For each CMOS test, 10 native speakers compare the samples generated by the two different models and give feedback with scores from -3 to +3. For each MOS test, 25 native speakers give their judgements of the overall performance with a 5-point scale. We evaluate model efficiency on two dimensions, inference time and memory cost. The inference process is performed on 1 NVIDIA Tesla K80 GPU. Long input for stress test is paragraph scripts from News domain.

4.1. Datasets

We first perform experiments on an en-US female voice (F001, $\sim$20h) in the News domain. Experiments presented from Session 4.3 to 4.5 are all based on this speaker's voice. We then expand the quality evaluation in Session 4.6 on a much more expressive voice: a zh-CN male voice (M002, $\sim$2h) with multi-style data recorded in the audiobook domain. The data of M002 contains 10 different styles/emotions, e.g., happy, angry, sad, etc. Other two zh-CN female voices (F003, F004), with the similar data distributions as M002, are in the multi-speaker and multi-style experiments. The test set of multi-style voices consists of 50 sentences, 5 for each style.

4.2. Model Configuration

4.2.1. Teacher: Autoregressive Transformer TTS model

According to model configurations in [8], We build a teacher model, autoregressive Transformer TTS model, for each voice. The teacher model is used for guiding the training of duration predictor in FastSpeech model and providing knowledge distillated mel-spectrograms. For voice F001, the teacher model is trained from scratch. But for the 3 zh-CN multi-style voices (M001, F003 and F004) with limited data resource, the teacher model is adapted from a robust source model, which is trained by $\sim$150h data from 13 speakers. The way of including speaker and style information into Transformer TTS model is similar as the illustration in Fig 1-(a) without the concatenation with decoder input.

4.2.2. Baseline FS

Our FastSpeech model consists of 4 FFT blocks on phoneme side and 6 FFT blocks on the mel-spectrogram side. The FFT block is made up by a 2-head self-attention and a 2-layer convolutional FFN. The $model_{dim}$ is set as 384 for both the dimension of phoneme embeddings and the hidden size of the self-attention. The ffn_{dim} in FFN block is set as 1536. For multi-speaker and multi-style model, the dimension of speaker / style embedding is set as 128.

4.2.3. ProbSparseFS

ProbSparseFS use ProbSparse self-attention described in Session 3.1 to replace the standard one in each FFT block. To set a proper information bandwidth of ProbSparse self-attention, the constant sampling factor c is tuned from {3,5,8,10}. We first use objective metrics to do a initial selection, which is a set of correlation with recordings: pitch, intensity, duration and pause. The overall distribution is similar among different value of c. We choose $c = 5$ (better in longer utterance) and $c = 10$ (better in short utterance) to do the CMOS test. The value of c is set as 10 in following experiments since the mean score is 0.017.

4.2.4. LinearizedFS

LinearizedFS is another efficient self-attention based FastSpeech model we proposed in this paper, which uses Linearized self-attention mentioned in Session 3.2. Theoretically, Linearized self-attention assumes a linear computational and memory complexity $O(N)$, which outperforms that of ProbSparseFS, $O(NlnN)$. This assumption is also verified by the following experiments analysis. Therefore, we further improve the efficiency of LinearizedFS by using a lightweight FFN block. We adjust the ffn_{dim} from 1536 to 768 and 512. Details are presented in the Session 4.5.

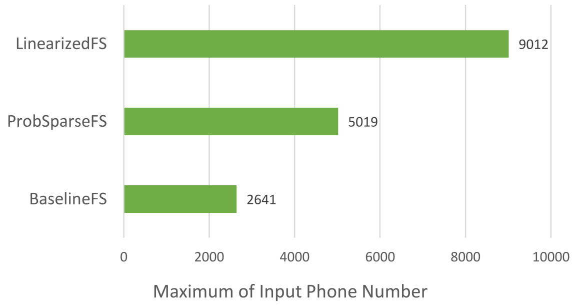

Fig. 2. Stress Test on Memory.

Table 1. CMOS test results (Mean score) for proposed model compared with BaselineFS on different Input Length.

Input Phone Number	748	1299	2072	2641
ProbSparseFS	-0.007	0.013	0.033	-0.018
LinearizedFS	0.017	0.067	0.089	0.072

4.3. Stress Test on Memory

We present this stress test result at first since it indicates the maximum input sequence length of each model. Fig 2 shows that the longest input length the BaselineFS model can tolerate is around 2.6k, which is about 16 general sentences. When input phone number exceeds the maximum, the inference process will be out-of-memory. ProbSparseFS shows a better tolerance to longer input length (∼5k), compared with BaselineFS. LinearizedFS has the lowest memory cost since the maximum input number is ∼9k, which is 1.8x of ProbSparseFS model and 3.4x of BaselineFS model.

4.4. Quality Test on Different Input Length

Based on the results of Fig 2, we conduct quality evaluation on test sets with different input length within the maximum input number of BaselineFS model. Table 1 shows the CMOS result of comparing proposed models with BaselineFS model. The input phone length is corresponding to 4/8/12/16 sentences, respectively. All results are within -0.1 to 0.1, which means the quality of proposed models is almost on-par with baseline model on different input lengths. Additionally, all CMOS scores of LinearizedFS model are positive and shows a growing tendency as the increase of input length, but some negative values appeared in ProbSparseFS' result. Therefore, further experiments in the following sessions will focus on LinearizedFS since it has advantage both in memory efficient (Session 4.3) and voice quality.

4.5. Performance Test on Inference Time

We further optimize FFN block with two lightweight FFN based LinearizedFS models, LinearizedFS-FFN768 and LinearizedFS-FFN512. Table 2 compares the inference time among different models. From the results, the time cost of LinearizedFS is obviously less than that of BaselineFS and the acceleration increases along with the input length. When the input phone number is 2641 (16 sentences), the speed-up brought by LinearizedFS is x2.12. The lightweight FFN accelerates the inference process in further by x2.96 (FFN768) and x3.61 (FFN512) with the same input.

Table 2. Performance Test on Time (ms).

Input Phone Number	748	1299	2072	2641
BaselineFS	387.9	728.4	1352.8	1961.6
LinearizedFS	258.1	433.7	789.4	926.6
	(x1.50)	(x1.68)	(x1.71)	**(x2.12)**
LinearizedFS	185.7	305.5	489.1	663.7
-FFN768	(x2.09)	(x2.38)	(x2.77)	**(x2.96)**
LinearizedFS	146.7	237.6	403	543.6
-FFN512	(x2.64)	(x3.07)	(x3.36)	**(x3.61)**

Table 3. MOS result on zh-CN expressive Voice.

AverageScore (CI-95%)	Recordings	BaselineFS	LinearizedFS -FFN512
(1) Multi-Style Models	4.36 (0.07)	4.21 (0.07)	4.24 (0.07)
(2) Multi-Speaker & Style Models	4.36 (0.08)	4.25 (0.07)	4.21 (0.07)

4.6. Extended Quality Test on Expressive Voice

Experiments in previous sessions 4.3 to 4.5 are tested on F001 voice, where focused on en-US News domain with a single speaking style. In this session, we conduct experiments on a more expressive zh-CN voice, M002, with multi-style data recorded in audiobook application domain. In the MOS test, we directly use LinearizedFS-FFN512 instead of LinearizedFS model to evaluate the voice quality of the most efficient model. Table 3-(1) shows the MOS result for single-speaker, multi-style based models. The MOS score of LinearizedFS-FFN512 is close to that of BaselineFS. The gap with recordings is 0.12, only slightly worse, and demonstrate good voice quality of our model. We also evaluate the voice quality of multi-speaker and multi-style based models in further for practical purposes. The MOS result is presented on Table 3-(2), showing consistent trend as Table 3-(1).

5. CONCLUSION AND FUTURE WORK

We proposed two improved FastSpeech TTS models, ProbSparseFS and LinearizedFS, whose self-attention and feed-forward network can make them memory efficient and computation fast, in comparing with the baseline FastSpeech. Their memory efficiencies are improved by 1.9 and 3.4 times, respectively, over the baseline. Additionally, LinearizedFS can yield a much faster inference speed than the baseline by 2.1 times. An improved lightweight FFN based LinearizedFS accelerates the inference by 3.6 times. All the new models can maintain good voice quality, almost on-par with that of the baseline, tested in en-US (English) News report and zh-CN (Mandarin) audiobook applications. The result is impressive and shows more exploration probability in future work. For example, based on the proposed model with better tolerance to input length, we can involve longer text, e.g., paragraph data, to incorporate more contextual information in training for further improvement of TTS voice quality.

6. REFERENCES

[1] S. Ö. Arık, M. Chrzanowski, A. Coates, G. Diamos, A. Gibiansky, Y. Kang, X. Li, J. Miller, A. Ng, J. Raiman, et al. Deep voice: Real-time neural text-to-speech. In *ICML*, pages 195–204. PMLR, 2017.

[2] R. Child, S. Gray, A. Radford, and I. Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

[3] 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. *NIPS*, 30, 2017.

[4] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. Oord, S. Dieleman, and K. Kavukcuoglu. Efficient neural audio synthesis. In *International Conference on Machine Learning*, pages 2410–2419. PMLR, 2018.

[5] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *ICML*, pages 5156–5165. PMLR, 2020.

[6] N. Kitaev, L. Kaiser, and A. Levskaya. Reformer: The efficient transformer. *ICLR*, 2020.

[7] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brébisson, Y. Bengio, and A. Courville. Melgan: Generative adversarial networks for conditional waveform synthesis. *arXiv preprint arXiv:1910.06711*, 2019.

[8] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu. Neural speech synthesis with transformer network. In *AAAI*, volume 33, pages 6706–6713, 2019.

[9] A. Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. Driessche, E. Lockhart, L. Cobo, F. Stimberg, et al. Parallel wavenet: Fast high-fidelity speech synthesis. In *ICML*, pages 3918–3926. PMLR, 2018.

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

[11] R. Prenger, R. Valle, and B. Catanzaro. Waveglow: A flow-based generative network for speech synthesis. In *ICASSP*, pages 3617–3621. IEEE, 2019.

[12] J. Qiu, H. Ma, O. Levy, S. W.-t. Yih, S. Wang, and J. Tang. Blockwise self-attention for long document understanding. *EMNLP*, 2019.

[13] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu. Fastspeech: Fast, robust and controllable text to speech. *arXiv preprint arXiv:1905.09263*, 2019.

[14] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *ICASSP*, pages 4779–4783. IEEE, 2018.

[15] Y. Tay, M. Dehghani, D. Bahri, and D. Metzler. Efficient transformers: A survey. *arXiv preprint arXiv:2009.06732*, 2020.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017.

[17] S. Wang, B. Z. Li, M. Khabsa, H. Fang, and H. Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

[18] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, et al. Tacotron: Towards end-to-end speech synthesis. *arXiv preprint arXiv:1703.10135*, 2017.

[19] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *AAAI*, 2021.