2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

TM-SPEECH: END-TO-END TEXT TO SPEECH BASED ON INTEGRATING TRANSFORMER AND MAMBA

Long Wang [1], Zichao Deng[1], Haoke Hou[1], Song Shen[1], Wancheng He[1]

and Haohan Ding [1]

[1] School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi 214122, China
wanglong@stu.jiangnan.edu.cn

Abstract. Text-to-speech (TTS) synthesis converts natural language text into speech, with deep learning driving recent advancements. While Transformer models excel at capturing dependencies, their quadratic complexity leads to high training costs. State-space models (SSMs), such as Mamba, offer sub-quadratic complexity, efficiently modeling long-range dependencies. We propose TM-Speech, a hybrid TTS system integrating Mamba for long-range dependencies and Transformer for short-range dependencies, reducing training costs while maintaining quality. Comparative experiments on LJSpeech and AISHELL-3 datasets show TM-Speech is 2× smaller and 3× faster than FastSpeech2 during training. Comprehensive evaluations using subjective metrics (Mean Opinion Score, Comparative Mean Opinion Score) and objective metrics (Word Error Rate, Character Error Rate) demonstrate TM-Speech achieves superior audio quality and intelligibility, with lower WER (4.65% vs. 5.12%) and CER (2.08% vs. 2.34%) on LJSpeech. The code is available at https://github.com/Apolarity886/TM-Speech.

Keywords: Text-to-speech, Speech Synthesis, Transformer, Mamba.

1 Introduction

Deep generative networks have been employed to synthesize understandable natural speech from text [1]. These networks have evolved from basic generative models, such as autoregressive models, to more advanced models, including Variational Autoencoders (VAE), Generative Adversarial Networks (GAN), normalizing flows, and diffusion models [2,3,4,5]. They have also transitioned from CNN/RNN-based architectures [6,7,8,9] to Transformer-based models [10]. However, CNN-based models are constrained by their local receptive fields, which significantly limits their ability to capture long-range dependencies and results in suboptimal performance in generation tasks. While Transformer-based models excel in global modeling, their self-attention mechanism exhibits quadratic complexity relative to the token length. This means that, when predicting the next token, the Transformer must consider relationships across all previous tokens in the sequence. Recent research indicates that state-space models (SSMs) have shown promising advancements in sequence modeling problems [11,12,13,14]. Among SSM-based models, Mamba [15] demonstrates performance comparable to Transformers in language modeling while maintaining linear time complexity.

Recent research has demonstrated that state-space models (SSMs) and Transformers are complementary in language modeling [16,17]. Based on this insight, we propose a hybrid architecture, TransMamba, built upon FastSpeech2 [18]. This architecture integrates the strengths of both Transformer and Mamba models for effective long-term and short-term sequence modeling, while also maintaining the advantage of fast inference and direct generation of audio waveforms from text.

We conducted combined experiments using the encoder-decoder architecture [2]. The results revealed that the hybrid architecture, which integrates both Transformer and Mamba, outperformed other configurations: the encoder-decoder with only Transformer modules, the encoder-decoder with only Mamba modules, and the Mamba encoder with Transformer decoder method. Consequently, we propose TM-Speech, an encoder-decoder neural network built using TransMamba modules. Experiments showed that TM-Speech achieved a 3× reduction in training time, while having only 50% of the model size compared to FastSpeech2. In addition, TM-Speech has lower WER and CER on LJSpeech.

2 Related Work

Transformer models, with their self-attention mechanisms, have become a cornerstone of speech processing, driving advancements in both automatic speech recognition (ASR) and text-to-speech (TTS) systems [19,20]. Models like FastSpeech2 [18] and VALL-E [23] leverage Transformers to achieve high-quality speech synthesis by modeling complex dependencies in sequential data. FastSpeech2, for instance, uses a non-autoregressive architecture with variance adaptors to predict duration, pitch, and energy, enabling faster and more controllable speech generation compared to autoregressive models. Similarly, VALL-E employs a language model-like approach to generate diverse speech outputs, excelling in zero-shot TTS scenarios. However, the quadratic complexity of Transformer attention mechanisms poses significant challenges, particularly for long-form speech, where memory and computational costs scale poorly with sequence length.
To address these limitations, state-space models (SSMs) have emerged as a compelling alternative due to their linear complexity and ability to model long-range dependencies efficiently. SSMs, such as Mamba [15], use structured state transitions to process sequences, offering memory-efficient computation compared to Transformers. In the TTS domain, recent studies have explored integrating SSMs with existing architectures to improve efficiency without sacrificing quality. For example, Jiang et al. [24] incorporated the Mamba module into VALL-E, replacing parts of the Transformer-based decoder with Mamba's state-space layers. Their results showed improved memory efficiency and faster inference times, though speech quality slightly lagged behind

Transformer-only models in certain expressive scenarios. Similarly, Miyazaki et al. [21] proposed VMamba, a FastSpeech2-based model that integrates bidirectional feed-forward Mamba modules into the variance adaptor and encoder-decoder framework. VMamba demonstrated superior speech generation speed and reduced memory footprint compared to FastSpeech2, particularly for long utterances, while maintaining comparable Mean Opinion Scores (MOS) in subjective evaluations.
Other works have explored hybrid architectures combining Transformers and SSMs to leverage their complementary strengths. For instance, Chen et al. [26] introduced a hybrid Transformer-Mamba model for ASR, where Mamba layers handled low-level feature extraction, and Transformer layers focused on high-level contextual modeling. Their approach achieved lower Word Error Rates (WER) than Transformer-only models on long-form audio datasets, highlighting Mamba's potential for scalable speech processing.

3 Model Description

In this section, we first describe the design motivation behind TM-Speech and then introduce the architecture of our model. TM-Speech leverages Mamba's exceptional capability in modeling one-dimensional sequences to enhance audio reasoning. This approach aims to simplify the training pipeline and improve speech quality for fully end-to-end text-to-waveform synthesis.

3.1 Motivation

FastSpeech2, which is based solely on Transformers, has achieved outstanding audio synthesis quality in the field of text-to-speech (TTS) [18]. However, due to the quadratic complexity of the attention mechanism, the encoder-decoder architecture composed of multi-layer feed-forward Transformer (FFTr) modules [26] requires a long time to converge when trained on large datasets and is highly dependent on GPU computing resources. Recently, state-space models (SSMs), particularly Mamba [15], have demonstrated excellent performance in modeling long sequences. This led us to consider incorporating the Mamba method into TTS, with the aim of reducing model training costs and improving inference audio quality. In the following subsection, we introduce the detailed design of TM-Speech, which is intended to address these challenges.

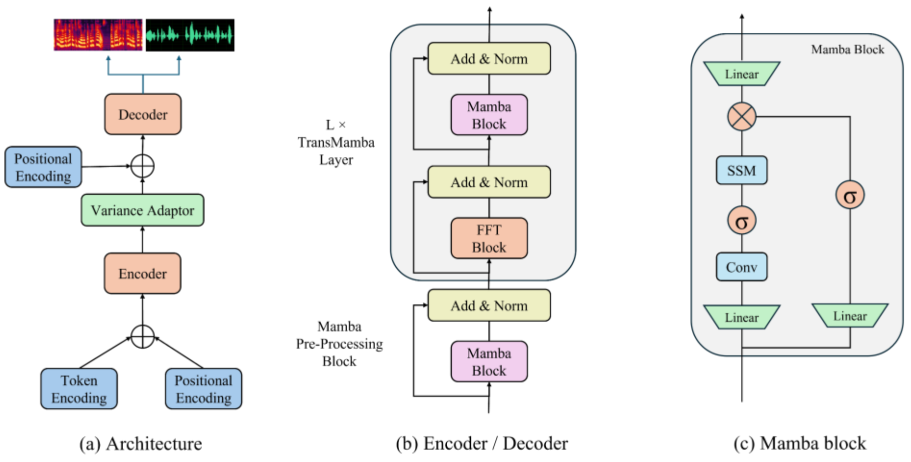

Fig. 1. The overall architecture for TM-Speech. Norm in subfigure (b) denotes RMSNorm in Mamba. SS1D in subfigure (c) denotes bidirectional scanning SSM module.

3.2 Model Overview

The architecture of TM-Speech is illustrated in Fig.1(a). It is built upon the FastSpeech2's encoder-decoder framework and primarily consists of four TransMamba stacks. As depicted in Fig.1(b), each TransMamba stack integrates a FFTr module with a Mamba module to model one-dimensional sequences. The process begins with the encoder, which adds the phoneme embedding sequence and positional encoding. This sequence first passes through the Mamba pre-processing module, then progresses through the four-layer TransMamba stack for encoding, and ultimately produces the phoneme hidden sequence.
Following the approach of FastSpeech2, we incorporate three types of variance information—duration, pitch, and energy—into the hidden sequence using a variance adapter. This augmentation enriches the hidden sequence with additional audio features. Finally, the decoder, which mirrors the structure of the encoder, converts the adapted hidden sequence into a mel spectrogram sequence in parallel.
Formally, let $X \in \mathbb{R}^{B\times L}$ denote input sequences with batch size B and length L and position encoding flow into the embedding layer. The embedding layer can be expressed as follows:

$$E = \mathrm{E}_{token}(X) \tag{1}$$

where $E \in \mathbb{R}^{B\times L\times D}$ is output embedding, D is the dimension of the embedding, E_{token} denote toke embedding layer. After that, we preprocess the sequence by a Mamba block to internally embed order information of input tokens. Mamba can be regarded as a RNN where the hidden state h_t at current time t is updated by the hidden state h_{t-1} at previous time t-1 as shown in the Equation 6.

$$\tilde{g} = \mathrm{TransMamba}\left(Mamba_{pre}(E)\right) \tag{7}$$

where the processed sequence $g \in \mathbb{R}^{B\times L\times D}$ flows into the TransMamba module to obtain the encoded sequence $\tilde{g} \in \mathbb{R}^{B\times L\times D}$. The Variance adaptor extracts the duration, pitch, and energy information from the sequence and compares them with the true value to perform mean square error (MSE).

$$\begin{aligned}\hat{d} &= \mathrm{DurationPredictor}(\tilde{g})\\ \hat{p} &= \mathrm{PitchPredictor}(\tilde{g})\\ \hat{e} &= \mathrm{EnergyPredictor}(\tilde{g})\end{aligned} \tag{3}$$

The predicted variational information is added to the sequence obtained by the encoder and then put into the decoder. The decoder converts the output sequence into a Mel spectrum sequence $\tilde{y} \in \mathbb{R}^{B\times L\times M}$ with Mel channel number M through a 1D convolutional layer.

$$\begin{aligned}\hat{g} &= \tilde{g} + \hat{p} + \hat{e}\\ \tilde{y} &= \mathrm{TransMamba}\left(Mamba_{pre}(\hat{g})\right)\\ \hat{y} &= \mathrm{Conv1D}(\tilde{y})\end{aligned} \tag{4}$$

We use HiFi-GAN as our vocoder to convert the mel spectrogram into the final audio. In the following subsections, we provide a detailed description of the TransMamba and the variance adapter designs.

3.3 TransMamba

As a key component of the model, TransMamba combines the advantages of both Transformer and Mamba, enabling the module to effectively handle both long- and short-range dependencies. Additionally, the linear complexity of Mamba significantly reduces the model's convergence time during training. TransMamba consists of a layer of basic FFTr modules from the Transformer and a layer of Mamba modules. The FFTr modules utilize a multi-head attention mechanism, allowing the model to focus on different parts of the sequence and capture contextual information.

As shown in Fig.3(c), the Mamba module is a sequence-to-sequence module with matching input and output dimensions. Specifically, Mamba takes the input and expands its dimensions through two linear projections. For one of these projections, Mamba processes the expanded embeddings using convolution and SiLU activation before passing them to the SS1D module. The core discrete 1D-Selective-Scan (SS1D) module, derived from the SSM component in Mamba, selects relevant information based on the input while filtering out irrelevant data.

The SSM is motivated by a continuous system that transforms an input function or sequence $x(t) \in \mathbb{R}$ into an output function or sequence $y(t) \in \mathbb{R}$ via an implicit latent state $h(t) \in \mathbb{R}^N$, defined as follows:

$$\begin{aligned}h'(t) &= Ah(t) + Bx(t)\\ y(t) &= Ch(t)\end{aligned} \tag{5}$$

where $A \in \mathbb{R}^{N\times N}$, $B \in \mathbb{R}^{N\times 1}$, and $C \in \mathbb{R}^{1\times N}$ are learnable matrices. The SSM can be discretized, converting continuous signals into discrete sequences with a step size Δ. The discretized form is expressed as:

$$\begin{aligned} h_t &= \bar{A}h_{t-1} + \bar{B}x_t \\ y &= Ch_t \end{aligned} \qquad (6)$$

where the discrete parameters $(\bar{A}, \bar{B})$ are derived from the continuous parameters (Δ, A, B) through a discretization rule, such as the zero-order hold (ZOH) method. Specifically, $\bar{A} = exp(\Delta A)$ and $\bar{B} = exp(\Delta A)^{-1}(exp(\Delta A) - I) * \Delta B$. After discretization, the model can be computed in two distinct ways: either as a linear recurrence for inference, as in Equation 5, or as a global convolution during training, as expressed in Equation 6:

$$\begin{aligned} \bar{K} &= (C\bar{B}, C\overline{AB}, \ldots, C\bar{A}^k\bar{B}) \\ y &= x \times \bar{K} \end{aligned} \qquad (7)$$

where $\bar{K}$ represents the convolutional kernel.
We enhance the SSM by introducing a bidirectional SS1D module. Specifically, we reverse the input sequence and process it separately through the SSM. The outputs of these two sequences are then concatenated and passed through a 2D convolutional network. This approach allows the module to capture bidirectional one-dimensional sequence information. The final output is residually connected to another projection after SiLU activation and combined with the output of the SS1D module through a multiplication gate. Ultimately, Mamba produces the final output via a linear projection.

3.4 Variance Adaptor

Similar to FastSpeech2, we use a variance adapter (see Fig.2(a)) to integrate duration, pitch, and energy information into the phoneme hidden sequence. During training, ground-truth values for these features guide the prediction of the target voice, while also serving as targets for training predictors for these features.
As shown in Fig.2(b), the predictors for duration, pitch, and energy have similar structures but different parameters. Unlike FastSpeech2, our ground-truth values for pitch and energy often include significant negative values. To handle potential information loss from negative values during ReLU activation, we use a 2-layer 1D convolutional network with Leaky ReLU, followed by normalization, dropout layers, and a final linear layer.

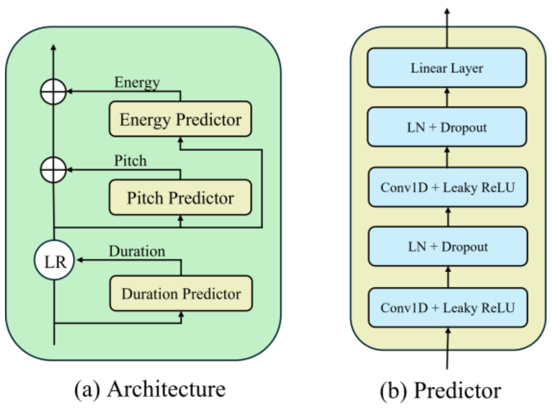

Fig. 2. The subfigure(a) shows the structure of the variance adaptor, where LR represents a module that adjusts the length of the encoder output according to the input duration information to match the length of the target spectrogram. The subfigure(b) illustrates the detailed structure of the duration, pitch, and energy predictors.

4 Experiments

The source code, along with pre-trained checkpoints and synthesized samples, is available online[1]. For evaluation purposes, we synthesize waveforms using a pre-trained HiFi-GAN.

4.1 Datasets and Model Configuration

We evaluate our TM-Speech model using two publicly available datasets: LJSpeech 1.1 [27] for English and AISHELL-3 [28] for Chinese. LJSpeech comprises 13,100 short audio clips from a single speaker reading passages from seven non-fiction books, recorded at 22,050 Hz. AISHELL-3 is a large-scale, high-fidelity Mandarin corpus with approximately 85 hours of recordings from 218 speakers, totaling 88,035 utterances. We split both datasets into 90% for training, 5% for validation, and 5% for testing, and select 100 samples from the test set for subjective evaluation.
Our TM-Speech model features a Mamba pre-processing block and four TransMamba blocks each in the encoder and decoder. The decoder's 1D-convolutional layer converts hidden states into 80-dimensional mel-spectrograms. The model is optimized using Smooth L1 loss, with phoneme embeddings and self-attention hidden sizes set to 256. Attention heads are configured as 4 in the encoder and 2 in the decoder.

[1] https://github.com/Apolarity886/TM-Speech

4.2 Evaluation

To provide a comprehensive assessment of TM-Speech, we evaluate both subjective and objective metrics. Subjective metrics include Mean Opinion Score (MOS) [29] for overall speech quality and Comparative Mean Opinion Score (CMOS) [30] to compare voice quality between models. Objective metrics include Word Error Rate (WER) and Character Error Rate (CER) to assess transcription accuracy and intelligibility, aligning with standard practices in Text-to-Speech (TTS) evaluation. We compare TM-Speech with FastSpeech2 and a Mamba-FFTr baseline, focusing on model size, training time, audio quality, and transcription accuracy.
Training and inference latency tests are conducted on a server with 32 GB of memory and an NVIDIA V100 GPU, using a batch size of 16 for training and 1 for inference. The Mamba module is configured with 8 SSID layers to match the 4-layer TransMamba module in TM-Speech.

4.2.1 Subjective Evaluation.

We conducted subjective evaluations with 30 native English speakers for LJSpeech and 20 native Chinese speakers for AISHELL-3. Participants assessed 100 synthesized samples from each model, rating overall quality (MOS) on a 1–5 scale and relative quality (CMOS) compared to FastSpeech2. Ground-truth (GT) recordings were included as a reference. Results are shown in Table 1.

Table 1. Reconstructed audio quality MOS of different methods on LJSpeech 1.1 and AISHELL-3, and inferred audio quality CMOS.

Dataset	Method	GT	MOS	CMOS
LJSpeech	FastSpeech 2	4.51±0.07	4.16±0.08	0.000
	Mamba-FFTr		4.27±0.09	+0.125
	TM-Speech		4.38±0.06	+0.274
AISHELL-3	FastSpeech 2	4.32±0.09	4.14±0.07	0.000
	Mamba-FFTr		3.91±0.08	-0.072
	TM-Speech		4.08±0.07	+0.118

TM-Speech achieves higher MOS and CMOS than FastSpeech2 on LJSpeech, indicating superior perceived quality and voice similarity. On AISHELL-3, TM-Speech slightly outperforms FastSpeech2 in CMOS, though its MOS is marginally lower, possibly due to challenges in modeling Mandarin prosody. Mamba-FFTr shows inconsistent performance, particularly on AISHELL-3, where its MOS is lower than both FastSpeech2 and TM-Speech.

4.2.2 Objective Evaluation.

To complement subjective metrics, we evaluate transcription accuracy using WER and CER, which measure the intelligibility of synthesized speech by comparing ASR-transcribed text to ground-truth transcriptions. We used the Whisper-large model [31] for transcription, with text normalization (e.g., removing punctuation, standardizing case) to ensure fair comparisons. WER is prioritized for LJSpeech due to its word-based structure, while CER is emphasized for AISHELL-3 given Mandarin's character-based nature.

Table 2. Table 3. Objective evaluation metrics (WER and CER) on LJSpeech 1.1 and AISHELL-3.

Dataset	Method	WER (%)	CER (%)
LJSpeech	FastSpeech 2	5.21	2.34
	Mamba-FFTr	4.87	2.19
	TM-Speech	**4.65**	**2.08**
AISHELL-3	FastSpeech 2	6.45	3.17
	Mamba-FFTr	7.23	3.56
	TM-Speech	**6.32**	**3.05**

TM-Speech achieves lower WER and CER than FastSpeech2 on both datasets, indicating improved intelligibility and transcription accuracy. On LJSpeech, TM-Speech's WER of 4.65% and CER of 2.08% outperform FastSpeech2 (5.12% and 2.34%) and Mamba-FFTr (4.87% and 2.19%), suggesting robust word- and character-level accuracy for English. On AISHELL-3, TM-Speech's CER of 3.05% is lower than FastSpeech2's 3.17%, though its WER is slightly better (6.32% vs. 6.45%). Mamba-FFTr performs worse on AISHELL-3, likely due to its reduced capacity to model Mandarin's tonal complexity. These results complement the subjective findings, confirming TM-Speech's ability to produce intelligible speech with fewer parameters.

4.2.3 Model Efficiency and Loss Metrics.

We compare model efficiency in Table 3, focusing on parameter count and training time. TM-Speech reduces both compared to FastSpeech2, leveraging the TransMamba fusion module.

Table 3. Model performance comparison. The Mamba in the method refers to the SSID module described in Figure 1b, and the FFTr refers to the feed-forward Transformer in Figure 1c.

Method	Params (M)	Training Time (h)
FastSpeech 2	35.21	21.21
Mamba-FFTr	23.56	**2.95**
TM-Speech	**18.09**	5.212

TM-Speech halves the parameters of FastSpeech2 (18.09M vs. 35.21M) and reduces training time on LJSpeech by 69% (5.212h vs. 21.21h). While Mamba-FFTr has the lowest training time (2.95h), its audio quality is less consistent (Table 1).

We also evaluate loss metrics in Table 4, including Mean Absolute Error (MAE) for duration, pitch, and energy, and Smooth L1 loss for mel-spectrograms.

Table 4. The mean absolute error (MAE) of the duration, pitch and energy in synthesized speech audio and Smooth L1 loss of mel between different methods in LJSpeech.

Method	Mel loss	Duration loss	Pitch loss	Energy loss
FastSpeech 2	0.412	**0.025**	**0.051**	**0.042**
Mamba-FFTr	0.259	0.075	0.392	0.204
TM-Speech	**0.171**	0.081	0.172	0.104

TM-Speech achieves the lowest Mel loss (0.171), indicating superior spectrogram restoration compared to FastSpeech2 (0.412) and Mamba-FFTr (0.259). However, its duration, pitch, and energy losses are higher than FastSpeech2, which may affect prosodic details in synthesized audio. These trade-offs are mitigated by TM-Speech's improved WER and CER, suggesting that intelligibility remains high despite less precise prosody modeling.

4.2.4 Discussion.

The inclusion of WER and CER provides a more comprehensive evaluation of TM-Speech, addressing the limitations of relying solely on subjective metrics like MOS and CMOS. WER and CER highlight TM-Speech's ability to produce intelligible speech, particularly for English (LJSpeech), where it outperforms FastSpeech2. For Mandarin (AISHELL-3), TM-Speech's CER advantage suggests robust character-level accuracy, though its WER is only marginally better, possibly due to challenges in modeling tonal variations. The Whisper-large ASR model used for transcription may introduce minor biases, particularly for Mandarin, where tone recognition remains a challenge. Future work could explore tone-specific metrics to further evaluate AISHELL-3 performance.

TM-Speech's efficiency (fewer parameters, reduced training time) and competitive performance across subjective and objective metrics make it a compelling alternative to FastSpeech2. However, its higher pitch and energy losses suggest room for improvement in prosody modeling, which could be addressed by refining the Trans-Mamba module or incorporating additional prosodic features.

5 Conclusion

We introduced TM-Speech, an efficient end-to-end Text-to-Speech (TTS) system combining Transformer and Mamba architectures. TM-Speech reduces model parameters by 49% (18.09M vs. 35.21M) and training time by 69% (5.212h vs. 21.21h) compared to FastSpeech2, while achieving comparable or better performance. Our enhanced evaluation, incorporating subjective metrics (MOS, CMOS) and objective metrics (WER, CER), demonstrates TM-Speech's superior audio quality and intelligibility on LJSpeech (MOS: 4.38 vs. 4.16; WER: 4.65% vs. 5.12%) and competitive results on AISHELL-3 (CMOS: +0.118; CER: 3.05% vs. 3.17%). However, higher pitch and energy losses indicate challenges in prosody modeling, particularly for Mandarin.

The addition of WER and CER strengthens our evaluation, aligning with TTS standards, though comparisons were limited to FastSpeech2, and ASR biases may affect Mandarin results. Future work will: (1) optimize prosody to reduce WER/CER, (2) compare TM-Speech with state-of-the-art TTS models, and (3) explore Mamba in generative TTS frameworks. TM-Speech offers a promising balance of efficiency and quality, paving the way for advancements in TTS research.

References

1. Tan X, Qin T, Soong F, et al. A survey on neural speech synthesis[J]. arXiv preprint arXiv:2106.15561, 2021.
2. Wang C, Chen S, Wu Y, et al. Neural codec language models are zero-shot text to speech synthesizers[J]. arXiv preprint arXiv:2301.02111, 2023.
3. Lee S, Ping W, Ginsburg B, et al. Bigvgan: A universal neural vocoder with large-scale training[J]. arXiv preprint arXiv:2206.04658, 2022.
4. Lee S H, Choi H Y, Lee S W. PeriodWave: Multi-Period Flow Matching for High-Fidelity Waveform Generation[J]. arXiv preprint arXiv:2408.07547, 2024.
5. Huang R, Lam M W Y, Wang J, et al. Fastdiff: A fast conditional diffusion model for high-quality speech synthesis[J]. arXiv preprint arXiv:2204.09934, 2022.
6. Van Den Oord A, Dieleman S, Zen H, et al. Wavenet: A generative model for raw audio[J]. arXiv preprint arXiv:1609.03499, 2016, 12.
7. Wang Y, Skerry-Ryan R J, Stanton D, et al. Tacotron: Towards end-to-end speech synthesis[J]. arXiv preprint arXiv:1703.10135, 2017.
8. Arık S Ö, Chrzanowski M, Coates A, et al. Deep voice: Real-time neural text-to-speech[C]//International conference on machine learning. PMLR, 2017: 195-204.
9. Tachibana H, Uenoyama K, Aihara S. Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention[C]//2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2018: 4784-4788.
10. Li N, Liu S, Liu Y, et al. Neural speech synthesis with transformer network[C]//Proceedings of the AAAI conference on artificial intelligence. 2019, 33(01): 6706-6713.
11. Quan C, Li X. Multichannel long-term streaming neural speech enhancement for static and moving speakers[J]. arXiv preprint arXiv:2403.07675, 2024.
12. Zhang X, Zhang Q, Liu H, et al. Mamba in Speech: Towards an Alternative to Self-Attention[J]. arXiv preprint arXiv:2405.12609, 2024.
13. Jiang X, Han C, Mesgarani N. Dual-path mamba: Short and long-term bidirectional selective structured state space models for speech separation[J]. arXiv preprint arXiv:2403.18257, 2024.
14. Waleffe R, Byeon W, Riach D, et al. An Empirical Study of Mamba-based Language Models[J]. arXiv preprint arXiv:2406.07887, 2024.
15. Gu A, Dao T. Mamba: Linear-time sequence modeling with selective state spaces[J]. arXiv preprint arXiv:2312.00752, 2023.
16. Lieber O, Lenz B, Bata H, et al. Jamba: A hybrid transformer-mamba language model[J]. arXiv preprint arXiv:2403.19887, 2024.
17. Park J, Park J, Xiong Z, et al. Can mamba learn how to learn? a comparative study on in-context learning tasks[J]. arXiv preprint arXiv:2402.04248, 2024.
18. Ren Y, Hu C, Tan X, et al. Fastspeech 2: Fast and high-quality end-to-end text to speech[J]. arXiv preprint arXiv:2006.04558, 2020.

19. Gulati A, Qin J, Chiu C C, et al. Conformer: Convolution-augmented transformer for speech recognition[J]. arXiv preprint arXiv:2005.08100, 2020.
20. Kim K, Wu F, Peng Y, et al. E-branchformer: Branchformer with enhanced merging for speech recognition[C]//2022 IEEE Spoken Language Technology Workshop (SLT). IEEE, 2023: 84-91.
21. Miyazaki K, Masuyama Y, Murata M. Exploring the Capability of Mamba in Speech Applications[J]. arXiv preprint arXiv:2406.16808, 2024.
22. Shams S, Dindar S S, Jiang X, et al. Ssamba: Self-supervised audio representation learning with mamba state space model[J]. arXiv preprint arXiv:2405.11831, 2024.
23. Erol M H, Senocak A, Feng J, et al. Audio Mamba: Bidirectional State Space Model for Audio Representation Learning[J]. arXiv preprint arXiv:2406.03344, 2024.
24. Jiang X, Li Y A, Florea A N, et al. Speech Slytherin: Examining the Performance and Efficiency of Mamba for Speech Separation, Recognition, and Synthesis[J]. arXiv preprint arXiv:2407.09732, 2024.
25. Zhu L, Liao B, Zhang Q, et al. Vision mamba: Efficient visual representation learning with bidirectional state space model[J]. arXiv preprint arXiv:2401.09417, 2024.
26. Vaswani A. Attention is all you need[J]. Advances in Neural Information Processing Systems, 2017.
27. Keith Ito. The lj speech dataset. https://keithito.com/LJ-Speech-Dataset/, 2017.
28. Shi Y, Bu H, Xu X, et al. Aishell-3: A multi-speaker mandarin tts corpus and the baselines[J]. arXiv preprint arXiv:2010.11567, 2020.
29. Chu M, Peng H. Objective measure for estimating mean opinion score of synthesized speech: U.S. Patent 7,024,362[P]. 2006-4-4.
30. Loizou P C. Speech quality assessment[M]//Multimedia analysis, processing and communications. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011: 623-654.