

MamTra: A Hybrid Mamba-Transformer Backbone for Speech Synthesis

Tan Dat Nguyen¹, Sangmin Bae¹, Joon Son Chung¹, Ji-Hoon Kim^{2,**}

¹ Korea Advanced Institute of Science and Technology, South Korea

² Chung-Ang University, South Korea

tandat.kaist@kaist.ac.kr

Abstract

Despite the remarkable quality of LLM-based text-to-speech systems, their reliance on autoregressive Transformers leads to quadratic computational complexity, which severely limits practical applications. Linear-time alternatives, notably Mamba, offer a potential remedy; however, they often sacrifice the global context essential for expressive synthesis. In this paper, we propose MamTra, an interleaved Mamba-Transformer framework designed to leverage the advantages of Mamba’s efficiency and Transformers’ modeling capability. We also introduce novel knowledge transfer strategies to distill insights from a pretrained Transformer into our hybrid architecture, thereby bypassing the prohibitive costs of training from scratch. Systematic experiments identify the optimal hybrid configuration, and demonstrate that MamTra reduces inference VRAM usage by up to 34% without compromising speech fidelity—even trained on only 2% of the original training dataset. Audio samples are available at¹.

Index Terms: text-to-speech, speech synthesis, hybrid architecture, state space model, mamba, MamTra

1. Introduction

The recent surge of Large Language Model (LLM)-based Text-to-Speech (TTS) systems has yielded near-human naturalness in expressive and multi-speaker speech generation [1, 2, 3, 4, 5, 6]. However, this success remains fundamentally dependent on autoregressive Transformer backbones. While their core self-attention mechanism effectively models global dependencies across the entire input sequences, it inherently incurs quadratic time and memory complexity with respect to sequence length [7]. In long-form synthesis scenarios, including podcasts, audiobooks, and streaming dialogue agents, this quadratic scaling leads to high latency and large key-value (KV) cache memory, making deployment impractical on edge devices [8, 9].

Efforts to mitigate the computational bottlenecks of self-attention have spanned both architectural modifications—such as grouped-query attention (GQA) [10], sparse attention [11], and pruning [12, 13]—and inference optimizations like KV cache compression [14, 15] and speculative decoding [16, 17]. While effective, these methods fundamentally remain optimizations atop the quadratic attention mechanism rather than structural replacements. Consequently, *linear* attention mechanisms have emerged as a scalable alternative computational primitive [18, 19, 20, 21, 22, 23]. In this context, State Space Models (SSMs)—most notably Mamba [20, 23]—have gained particular traction for their ability to combine linear-time

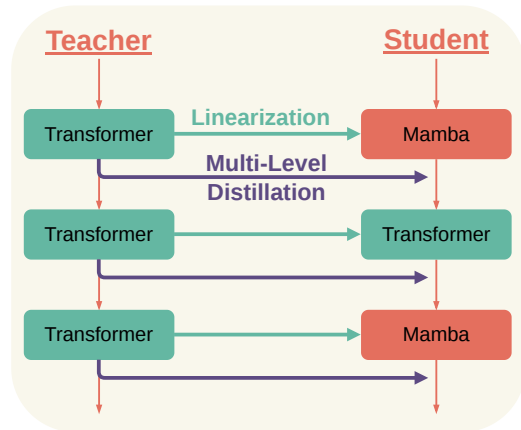


Figure 1: Overview of the hybrid Mamba-Transformer configurations for speech synthesis. Selective Transformer-to-Mamba layer transfer reduces training cost and accelerates convergence, while performance is recovered via knowledge distillation using less than 2% of the teacher’s English training data.

inference with strong local temporal modeling. This potential has driven recent adoption in a range of speech tasks, where Mamba-based architectures are demonstrating significant efficiency gains [24, 25, 8, 26, 27, 28, 29, 30]. However, this efficiency often comes at the cost of expressivity: pure SSMs still underperform Transformers at scale [31], particularly in in-context learning [32, 33], and long-context reasoning [34].

To bridge this gap, *hybrid* architectures have become a promising solution, strategically interleaving Mamba blocks with Transformer blocks. These designs combine the computational efficiency of Mamba for local acoustic modeling with the global semantic reasoning of Transformers [32, 35, 36, 37, 9]. However, within the TTS domain, such hybrid designs remain largely unexplored. The *only* existing attempt [38] relies on inefficient pretraining from scratch, and its technical implementation details have not been released. To prevent the high cost of pretraining, we leverage insights from recent model conversion techniques [39, 40, 41, 13]. Specifically, we hypothesize that a pre-trained Transformer can be transformed into a hybrid TTS model by replacing a subset of its blocks with Mamba layers via attention-to-SSM parameter transfer. This structured initialization ensures that the model retains teacher-level speech fidelity while achieving significantly faster inference.

Our contributions are summarized as follows: (1) We introduce **MamTra**, the first systematic study of hybrid Mamba-Transformer architecture for speech synthesis. It reuses pretrained Transformer weights to initialize both

** indicates the corresponding author.

¹Project page: <https://mm.kaist.ac.kr/projects/mamtra/>

Table 1: Architectural replacement strategies mapping Transformer (blue blocks) and Mamba (orange blocks) layers. *BlockBeg stands for BlockBeginning.

Category	Strategy	Structure
Interleaved	BlockBeg*	
	BlockEnd	
Contiguous	Front	
	Middle	
	Back	
	Sandwich	
Data-Driven	Importance	

retained Transformer layers and newly introduced Mamba blocks, enabling selective replacement without pretraining from scratch (Figure 1). (2) We conduct a speech-centric analysis of the hybrid design space, identifying where and how many to place Mamba layers and quantifying a clear efficiency–quality trade-off. Our results demonstrate a strong trade-off: using knowledge distillation on just 2% of the English data compared to baseline yields a 34% reduction in GPU memory with only a 0.25% absolute increase in WER while maintaining the naturalness and expressiveness of the teacher. (3) Finally, we validate MamTra’s efficiency through both extensive experiments and theoretical analysis, confirming sub-quadratic complexity and sub-linear KV-cache growth. This corresponds to a reduction of up to $1.4e^{11}$ FLOPs per token at a context length of 2,048.

2. MamTra Hybrid Architecture

2.1. Preliminaries

Transformer [7] has become the dominant backbone for sequence modeling due to its ability to capture long-range dependencies through self-attention mechanism. Each token attends to relevant context (e.g., preceding tokens in autoregressive models) by computing attention scores:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}, \quad (1)$$

where $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{L \times d}$ are the query, key, and value matrices for sequence length L and hidden dimension d . While caching the full KV history ensures precise global reasoning, its quadratic complexity $\mathcal{O}(L^2)$ severely bottlenecks long-form generation [7].

In contrast, **Mamba** [20, 23] builds upon SSMs, which map a 1D input sequence $x(t)$ to an output $y(t)$ through a latent state $h(t)$. Unlike traditional time-invariant SSMs, Mamba introduces *selectivity* by making the state parameters functions of the current input \mathbf{x}_t . The sequence is modeled through a compressed hidden state $\mathbf{h}_t \in \mathbb{R}^N$, updated *recurrently*:

$$\mathbf{h}_t = \bar{\mathbf{A}}_t \mathbf{h}_{t-1} + \bar{\mathbf{B}}_t \mathbf{x}_t, \quad \mathbf{y}_t = \mathbf{C}_t \mathbf{h}_t, \quad (2)$$

where $\bar{\mathbf{A}}_t$ and $\bar{\mathbf{B}}_t$ are obtained by discretizing the continuous system matrices via a zero-order hold (ZOH) [19] and the Euler method. By projecting the discretization parameters Δ_t , \mathbf{B}_t , and \mathbf{C}_t from the input \mathbf{x}_t , the model achieves a selective mechanism that dynamically propagates or forgets information based on the context [23]. Mamba leverages parallel scan algorithms [42, 43] for highly efficient parallel training, and ensures linear $\mathcal{O}(L)$ decoding time with constant $\mathcal{O}(1)$ memory footprint during inference.

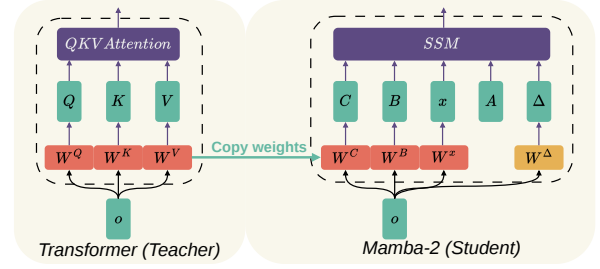


Figure 2: Following the alignment between Eq. 2 and Eq. 4, projection weights for C , B , and x in Mamba are initialized with Transformer’s Q , K , and V projection weights, respectively.

2.2. Hybridization Strategy

We explore various Transformer-to-Mamba replacement strategies to balance global context modeling with local temporal efficiency. As summarized in Table 1, we categorize these into four design spaces: (1) **Interleaved** (BlockBeg/End), preserving periodic Transformer layers; (2) **Contiguous** (Front/Middle/Back/Sandwich), grouping layer types spatially into continuous blocks or at the model’s boundaries; and (3) **Data-driven** (Importance) [40], replacing less critical layers based on cosine similarity or WER criteria. Each strategy is evaluated across Transformer-Mamba ratios ranging from 1:1 to an aggressive 1:11 to determine the optimal functional regions for each primitive.

2.3. Linearization and Structural Mapping

The transition from self-attention to the recurrent form can be interpreted as a linearization [24]. To bridge these architectures at timestep t , we apply a causal mask $m_{s,t} = 1$ for $s \leq t$ (0 otherwise). Removing the softmax allows using associativity to decouple the query from historical context:

$$\mathbf{y}_t = \frac{\mathbf{Q}_t^\top}{\sqrt{d_k}} \left(\sum_{s \leq t} m_{s,t} \mathbf{K}_s \mathbf{V}_s \right). \quad (3)$$

This formulation reveals an underlying recurrence where the accumulated product $\mathbf{h}_t = \sum_{s \leq t} m_{s,t} \mathbf{K}_s \mathbf{V}_s$ serves as the latent state. Utilizing the property $m_{s,t} = m_{t-1,t} \cdot m_{s,t-1}$ for causal sequences (i.e., enabling \mathbf{h}_t to be updated recursively from \mathbf{h}_{t-1}), the update rule is expressed in a linear RNN form:

$$\mathbf{h}_t = m_{t-1,t} \mathbf{h}_{t-1} + \mathbf{K}_t \mathbf{V}_t, \quad \mathbf{y}_t = \frac{1}{\sqrt{d_k}} \mathbf{Q}_t^\top \mathbf{h}_t. \quad (4)$$

Comparing Eq. 2 and Eq. 4 reveals a direct structural alignment. Specifically, the key-value product ($\mathbf{K}_t \mathbf{V}_t$) corresponds to the SSM’s input update ($\bar{\mathbf{B}}_t \mathbf{x}_t$), with the key (\mathbf{K}_t) mapping to the input-dependent parameter $\bar{\mathbf{B}}_t$ and the value (\mathbf{V}_t) to the projected input \mathbf{x}_t . The query (\mathbf{Q}_t) aligns perfectly with the output projection (\mathbf{C}_t). Leveraging this equivalence, the pretrained Transformer’s query, key, and value projection weights are directly transferred to initialize the corresponding Mamba layers for projections of \mathbf{C}_t , $\bar{\mathbf{B}}_t$, and \mathbf{x}_t (Figure 2).

2.4. Multi-Level Distillation

While the closed-form weight transfer provides a strong structural initialization, relying solely on these weights is insufficient to preserve the teacher’s performance. Specifically, removing the softmax nonlinearity fundamentally alters the

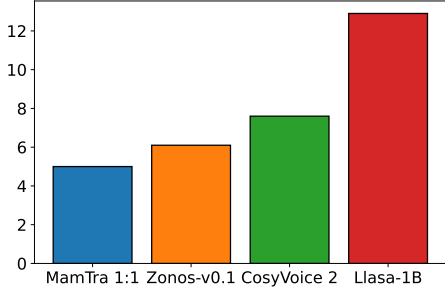


Figure 3: Memory usage (GB) on Seed-TTS-eval (using NVIDIA A6000). MamTra 1:1 reduces VRAM by 34% and 17% compared to CosyVoice 2 and Zonos-v0.1, respectively.

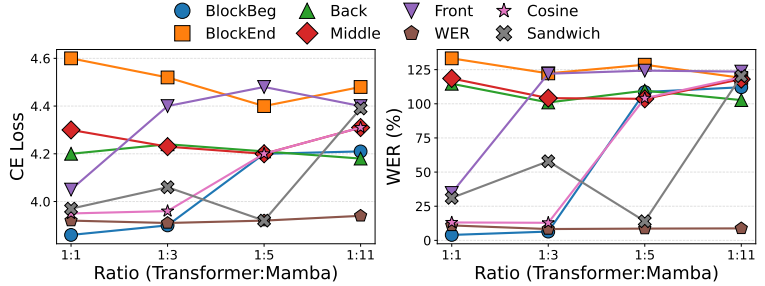


Figure 4: Cross-entropy (CE) loss and Word Error Rate (WER) on Seed-TTS-eval for hybrid Mamba-Transformer variants after 15 training epochs on LibriTTS (0.5k h). CE loss exhibits a consistent correlation with WER across different ratios and placement strategies.

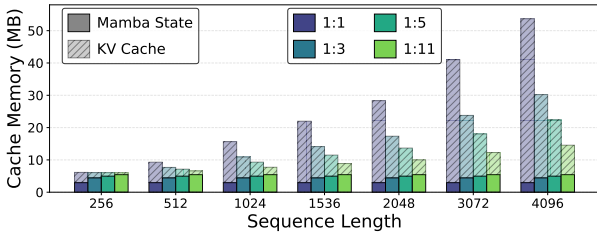


Figure 5: Cache size growth in the hybrid model, where the sequence-length dependency scales with the number of remaining Transformer layers.

attention dynamics, thereby reducing the model’s expressive capacity [40]. To bridge this performance gap, MamTra employs a multi-level distillation strategy designed to transfer both the representational structure and the generation behavior of the teacher Transformer into the hybrid student model. The overall training objective is defined as:

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \mathcal{L}_{\text{logits}} + \mathcal{L}_{\text{emb}}, \quad (5)$$

where \mathcal{L}_{CE} provides ground-truth output supervision via cross-entropy. To recover the generation behavior lost during linearization, $\mathcal{L}_{\text{logits}}$ minimizes the skew KL divergence [44] between the teacher’s and student’s logits. Finally, \mathcal{L}_{emb} enforces a mean-squared error constraint on the token embeddings per input sequence. Together, these objectives align the student’s representations with the teacher’s joint semantic-acoustic space, ensuring stable adaptation to the Mamba blocks despite the architectural substitution.

3. Experimental Setup

Datasets. We utilize the LibriTTS dataset [45] for model training, and the Seed-TTS-eval `test-en` benchmark [46] alongside the LibriTTS `test-clean` split for evaluation. While Seed-TTS-eval `test-en` spans a balanced target text length ranging from 3 to 27 words, we additionally construct a LibriTTS-based test set encompassing a significantly wider length range (1 to 62 words). This evaluates the robustness of MamTra under length-stressed conditions.

Baselines. We compare our approach against three strong baselines: (1) **CosyVoice 2** [6], built on the Qwen2.5 architecture [47]; (2) **Llasa-1B** [48], utilizing Llama-3.2 [49]; and (3) **Zonos-v0.1** [38], which represents the first hybrid Transformer-SSM model for TTS, although its precise architectural details remain largely unreported.

Table 2: Comparison of sequence-scaling computational and memory complexity across architectures. Here, L denotes sequence length, d the hidden dimension, d_{kv} the key-value dimension in GQA [10], and N the state size of Mamba.

Model	Compute	Memory	
	FLOPs	Training	Inference
MHA	$\mathcal{O}(L^2d + Ld^2)$	$\mathcal{O}(L^2 + Ld)$	$\mathcal{O}(Ld)$
GQA	$\mathcal{O}(L^2d + Ld^2)$	$\mathcal{O}(L^2 + Ld)$	$\mathcal{O}(Ld_{kv})$
Mamba	$\mathcal{O}(Ld^2)$	$\mathcal{O}(Ld)$	$\mathcal{O}(dN)$

Training Details. We conduct the performance recovery and fine-tuning of MamTra based on the CosyVoice 2 backbone [6]. All models are trained on NVIDIA A6000 GPUs using the Adam optimizer with a learning rate of $1e^{-5}$ and a dynamic batch size of 40,000 tokens. While most experimental configurations are trained for 15 epochs to analyze architectural trends, models for the final rigorous comparisons (Table 3) are trained for 50 epochs to ensure full convergence.

Evaluation Metrics. We comprehensively assess the models based on computational efficiency, alongside the intelligibility and perceptual quality. Computational efficiency is quantified using TFLOPs, GPU memory usage, and cache size. For intelligibility, we report the Word Error Rate (WER). Perceptual quality is assessed via Speaker Similarity (SSIM), UTMOS, and Naturalness Mean Opinion Score (NMOS). All objective metrics are computed utilizing the `versa` toolkit [50]. For the subjective NMOS evaluation, 50 utterances are randomly sampled per model and rated by 15 independent listeners.

4. Analysis

4.1. Efficiency and Complexity Analysis

The structural mapping between Attention and Selective SSMS yields significant scaling differences. As shown in Table 2, while GQA-based backbones like Llasa and CosyVoice 2 reduce cache footprint, they retain quadratic $\mathcal{O}(n^2d)$ complexity. Conversely, Mamba [20] achieves linear $\mathcal{O}(nd^2)$ scaling. For **MamTra**, this hybrid integration enables *sub-quadratic computation* and *sub-linear cache growth*, as complexity is dominated by the remaining Transformer layers. This is empirically confirmed in Figure 5, where Mamba state remains nearly constant while KV-cache growth is significantly attenuated relative to sequence length. In practice, MamTra reduces average inference-time memory by up to 34% compared to the CosyVoice 2 baseline, as demonstrated in Figure 3.

Table 3: Comparison of TTS Models Across Quality and Efficiency Metrics in Seed-TTS-eval test-en and LibriTTS test-clean. We compute the FLOPs needed for the LLM backbone to infer one token with a context length of 2048 (1 TFLOP = $1e^{12}$ FLOPs). All default cache enabled. NMOS is computed with confidence interval of 95%.

Model	Hybrid	Source	Params	Layers	Data (kh)	Ratio	Strategy	TFLOPs ↓	NMOS ↑	Seed-TTS-eval test-en			LibriTTS test-clean		
										UTMOS ↑	SSIM ↑	WER ↓	UTMOS ↑	SSIM ↑	WER ↓
Groundtruth	-	-	-	-	-	-	-	-	3.93 ± 0.15	3.52	1.00	1.47	4.14	1.00	1.85
Llasa-1B	×	Public	1.3B	16	250	1:0	-	4.53	3.64 ± 0.19	4.13	0.46	3.54	4.30	0.40	3.07
CosyVoice 2	×	Public	0.5B	24	170	1:0	-	1.78	3.68 ± 0.16	4.15	0.66	2.03	4.35	0.74	2.04
Zonos-v0.1	✓	Public	1.6B	46	200	1:3	BlockBeg	7.32	3.18 ± 0.18	3.63	0.67	3.42	3.96	0.72	3.37
MamTra	✓	Finetuned	0.5B	24	0.5	1:1	BlockBeg	1.64	3.66 ± 0.16	4.16	0.72	2.28	4.35	0.75	2.26
	✓	Finetuned	0.5B	24	0.5	1:3	BlockBeg	1.57	3.65 ± 0.16	4.14	0.72	3.26	4.35	0.75	2.99
	✓	Finetuned	0.5B	24	0.5	1:5	WER	1.54	3.65 ± 0.16	4.13	0.72	2.53	4.36	0.75	2.28
	✓	Finetuned	0.5B	24	0.5	1:11	WER	1.52	3.27 ± 0.18	4.15	0.72	3.99	4.37	0.75	4.08

Table 4: Impact of different loss components after 15 epochs.

Components	WER ↓	CER ↓	SSIM ↑	UTMOS
MamTra 1:1	3.48	2.86	0.72	4.16
w/o \mathcal{L}_{CE}	6.70	4.32	0.72	4.15
w/o \mathcal{L}_{logits}	6.13	3.46	0.72	4.15
w/o \mathcal{L}_{emb}	5.63	3.26	0.72	4.15

4.2. Efficiency-Quality Tradeoff

As shown in Figure 4, the BlockBeg scheme consistently achieves lower CE and WER across ratios, while alternative placements exhibit higher variance and reduced robustness as the Mamba ratio increases. Notably, as the degree of layer removal becomes more aggressive, the advantage of WER-based layer importance becomes increasingly pronounced, consistent with prior observations in structured pruning [13]. Based on these consistent findings under low-cost settings, we select the **BlockBeg** strategy as the replacement scheme for ratios of 1:1 and 1:3; **WER** strategy for ratios 1:5 and 1:11.

Table 3 illustrates MamTra’s quality–efficiency trade-off relative to Transformer-only and prior hybrid models. While the backbone network (**CosyVoice 2**) provides high quality, it incurs significant attention-based overhead. By replacing attention layers with Mamba blocks, MamTra reduces per-token FLOPs by up to 1.4×10^{11} at a 2048 context length while maintaining comparable NMOS, UTMOS, and speaker similarity. Specifically, the **MamTra 1:1** configuration preserves strong intelligibility, matching the teacher’s perceptual metrics with only a modest 0.25% absolute WER increase. Other variants, such as 1:3 and 1:5, offer further computational gains with manageable increases in word error rate. However, the most aggressive 1:11 ratio represents the upper bound of this trade-off, where intelligibility begins to degrade below the Zonos-v0.1 baseline. Furthermore, MamTra achieves up to a 34% reduction in average inference memory, consistent with the sub-linear cache scaling in Figure 5. These results demonstrate that MamTra effectively balances naturalness with the efficiency required for memory-constrained, long-context TTS deployment.

4.3. Sensitivity Under Length-Stress Conditions

In the length-stress condition (1–62 words), the MamTra 1:5 (WER) configuration demonstrates a remarkably stable efficiency-quality trade-off. Despite the aggressive substitution of attention layers for Mamba blocks, the importance-based WER selection strategy keeps the WER exceptionally low at 2.28%. Notably, this configuration exhibits superior robustness compared to the more conservative 1:3 (BlockBeg) variant

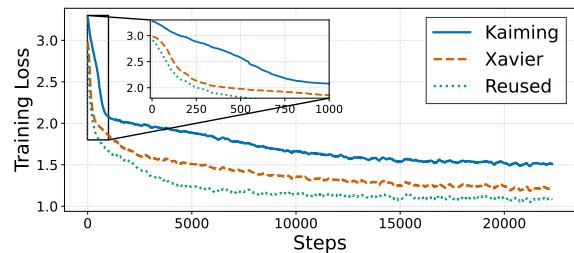


Figure 6: Convergence speed after 15 epochs.

(WER = 2.99%) and achieves parity with the 1:1 (BlockBeg) baseline while maintaining high UTMOS of 4.36 and SSIM of 0.75, consistently surpassing Zonos. This suggests that under length-stressed conditions, better layer placement is more effective than simply retaining more Transformer blocks.

4.4. Training Dynamics and Loss Contributions

Ablation studies summarized in Table 4 confirm that \mathcal{L}_{CE} , \mathcal{L}_{logits} , and \mathcal{L}_{emb} are essential for effective knowledge transfer, primarily impacting linguistic accuracy rather than perceptual quality. Removing ground-truth supervision (\mathcal{L}_{CE}) results in the most severe degradation in intelligibility, while the logit distillation loss (\mathcal{L}_{logits}) using Skew KL-Divergence is critical for recovering the teacher’s nuanced generation behavior.

Complementary to these objectives, the initialization strategy significantly impacts training efficiency. As illustrated in Figure 6, reusing pretrained Transformer weights for both attention and Mamba blocks facilitates much faster convergence and a lower final training loss compared to standard Xavier or Kaiming initialization. This “initialize-and-train” approach allows the hybrid model to maintain a clear performance gap from the outset, enabling successful performance recovery using less than 2% of the original teacher’s training data.

5. Conclusion

We proposed MamTra, a hybrid Mamba-Transformer framework designed for efficient and high-quality speech synthesis. Through rigorous analysis, we optimized the integration and introduced custom initialization with multi-level distillation. Our experimental results demonstrate that MamTra achieves up to a 34% reduction in inference memory and saves $1.4e^{11}$ FLOPs per token without compromising generation quality. These findings suggest that MamTra serves as a practical and scalable solution for memory-efficient TTS systems. We hope that our comprehensive analysis provides a solid foundation for future research in hybrid modeling for speech tasks.

6. Acknowledgement

This work was supported by the IITP-ITRC grant funded by the Korea government (Ministry of Science and ICT, IITP-2026-RS-2023-00259991).

7. Generative AI Use Disclosure

During the preparation of this manuscript, the authors used Google’s Gemini as a writing assistant for editing, language polishing, and structural refinement. This tool was used solely to improve the readability and overall clarity of the prose. For implementation-level assistance, the authors utilized xAI’s Grok for code refinement and debugging, and Anthropic’s Claude for code cleaning ahead of public release and for project page creation. These tools were not involved in the design of the proposed methods, experimental protocols, or scientific interpretations. All content, code, and conclusions were reviewed, revised, and approved by the authors, who take full responsibility for the integrity of the manuscript and its submission.

8. References

- [1] Z. Borsos, R. Marinier, D. Vincent *et al.*, “Audiolm: a language modeling approach to audio generation,” *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 31, pp. 2523–2533, 2023.
- [2] C. Wang, S. Chen, Y. Wu, Z. Zhang, L. Zhou, S. Liu, Z. Chen, Y. Liu, H. Wang, J. Li *et al.*, “Neural codec language models are zero-shot text to speech synthesizers,” *arXiv:2301.02111*, 2023.
- [3] Z. Du, Q. Chen, S. Zhang, K. Hu *et al.*, “Cosyvoice: A scalable multilingual zero-shot text-to-speech synthesizer based on supervised semantic tokens,” *arXiv:2407.05407*, 2024.
- [4] Y. Chen, Z. Niu, Z. Ma, K. Deng, C. Wang, J. JianZhao, K. Yu, and X. Chen, “F5-tts: A fairytaler that fakes fluent and faithful speech with flow matching,” in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025, pp. 6255–6271.
- [5] Y. Song, X. Zhuang, J. Chen, Z. Niu, G. Yang, C. Du, Z. Chen, Y. Wang, Y. Wang, and X. Chen, “DiSTAR: Diffusion over a scalable token autoregressive representation for speech generation,” *arXiv preprint arXiv:2510.12210*, 2025.
- [6] Z. Du, Y. Wang, Q. Chen, X. Shi, X. Lv *et al.*, “Cosyvoice 2: Scalable streaming speech synthesis with large language models,” *arXiv:2412.10117*, 2024.
- [7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, “Attention is all you need,” in *NeurIPS*, 2017.
- [8] S. J. Park, J. Salazar, A. Jansen, K. Kinoshita, Y. M. Ro, and R. Skerry-Ryan, “Long-form speech generation with spoken language models,” in *Proc. ICML*, 2025.
- [9] S. Bae, B. Acun, H. Habeeb, S. Kim, C.-Y. Lin, L. Luo, J. Wang, and C.-J. Wu, “Hybrid architectures for language models: Systematic analysis and design insights,” *arXiv:2510.04800*, 2025.
- [10] J. Ainslie, J. Lee-Thorp, M. De Jong, Y. Zemlyanskiy, F. Lebrón, and S. Sanghai, “Gqa: Training generalized multi-query transformer models from multi-head checkpoints,” *arXiv preprint arXiv:2305.13245*, 2023.
- [11] A. Liu, A. Mei, B. Lin, B. Xue, B. Wang, B. Xu, B. Wu, B. Zhang, C. Lin, C. Dong *et al.*, “Deepseek-v3. 2: Pushing the frontier of open large language models,” *arXiv preprint arXiv:2512.02556*, 2025.
- [12] X. Ma, G. Fang, and X. Wang, “Llm-pruner: On the structural pruning of large language models,” in *NeurIPS*, 2023.
- [13] T. D. Nguyen, J. Kim, J.-H. Kim, S. Choi, Y. Lim, and J. S. Chung, “Spade: Structured pruning and adaptive distillation for efficient llm-tts,” in *Proc. ICASSP*, 2026.
- [14] S. Ge, Y. Zhang, L. Liu, M. Zhang, J. Han, and J. Gao, “Model tells you what to discard: Adaptive KV cache compression for LLMs,” in *Workshop on Advancing Neural Network Training: Computational Efficiency, Scalability, and Resource Optimization (WANT@NeurIPS 2023)*, 2023.
- [15] Z. Zhang, Y. Sheng, T. Zhou, T. Chen, L. Zheng, R. Cai, Z. Song, Y. Tian, C. Ré, C. Barrett, Z. A. Wang, and B. Chen, “H2o: Heavy-hitter oracle for efficient generative inference of large language models,” in *Proc. ICML*, 2023.
- [16] T. D. Nguyen, J.-H. Kim, J. Choi, S. Choi, J. Park, Y. Lee, and J. S. Chung, “Accelerating codec-based speech synthesis with multi-token prediction and speculative decoding,” in *Proc. ICASSP*, 2025.
- [17] B. Li, H. Wang, S. Zhang, Y. Guo, and K. Yu, “Fast and high-quality auto-regressive speech synthesis via speculative decoding,” in *Proc. ICASSP*, 2025.
- [18] Y. Sun, L. Dong, S. Huang, S. Ma, Y. Xia, J. Xue, J. Wang, and F. Wei, “Retentive network: A successor to transformer for large language models,” *arXiv preprint arXiv:2307.08621*, 2023.
- [19] A. Gu, K. Goel, and C. Ré, “Efficiently modeling long sequences with structured state spaces,” in *Proc. ICLR*, 2022.
- [20] T. Dao and A. Gu, “Transformers are ssm: Generalized models and efficient algorithms through structured state space duality,” in *Proc. ICML*, 2024.
- [21] S. De, S. L. Smith, A. Fernando, A. Botev, G. Cristian-Muraru, A. Gu, R. Haroun, L. Berrada, Y. Chen, S. Srinivasan *et al.*, “Griffin: Mixing gated linear recurrences with local attention for efficient language models,” *arXiv preprint arXiv:2402.19427*, 2024.
- [22] S. Yang, J. Kautz, and A. Hatamizadeh, “Gated delta networks: Improving mamba2 with delta rule,” in *Proc. ICLR*, 2025.
- [23] A. Gu and T. Dao, “Mamba: Linear-time sequence modeling with selective state spaces,” in *First conference on language modeling*, 2024.
- [24] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret, “Transformers are rns: fast autoregressive transformers with linear attention,” in *Proc. ICML*, 2020.
- [25] D. Y. Fu, T. Dao, K. K. Saab, A. W. Thomas, A. Rudra, and C. Ré, “Hungry Hungry Hippos: Towards language modeling with state space models,” in *Proc. ICLR*, 2023.
- [26] S. Yang, B. Wang, Y. Shen, R. Panda, and Y. Kim, “Gated linear attention transformers with hardware-efficient training,” in *Proc. ICML*, 2024.
- [27] X. Jiang, Y. A. Li, A. N. Florea, C. Han, and N. Mesgarani, “Speech slytherin: Examining the performance and efficiency of mamba for speech separation, recognition, and synthesis,” in *Proc. ICASSP*, 2025.
- [28] X. Zhang, Q. Zhang, H. Liu, T. Xiao, X. Qian, B. Ahmed, E. Ambikairajah, H. Li, and J. Epps, “Mamba in speech: Towards an alternative to self-attention,” *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, 2025.
- [29] X. Gao and N. F. Chen, “Speech-mamba: Long-context speech recognition with selective state spaces models,” in *IEEE Spoken Language Technology workshop*. IEEE, 2024, pp. 1–8.
- [30] D. Kwak, Y. Jang, S. Kim, and J. S. Chung, “Ednet: A versatile speech enhancement framework with gating mamba mechanism and phase shift-invariant training,” *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, pp. 1200–1215, 2026.
- [31] S. Jelassi, D. Brandfonbrener, S. M. Kakade, and E. Malach, “Repeat after me: Transformers are better than state space models at copying,” in *Proc. ICML*, 2024.
- [32] J. Park, J. Park, Z. Xiong, N. Lee, J. Cho, S. Oymak, K. Lee, and D. Papailiopoulos, “Can mamba learn how to learn? a comparative study on in-context learning tasks,” in *Proc. ICML*, 2024.
- [33] R. Grazi, J. Siems, S. Schrodi, T. Brox, and F. Hutter, “Is mamba capable of in-context learning?” in *AutoML 2024 Methods Track*, 2024.

- [34] Z. Ye, K. Xia, Y. Fu, X. Dong, J. Hong, X. Yuan, S. Diao, J. Kautz, P. Molchanov, and Y. C. Lin, "LongMamba: Enhancing mamba's long context capabilities via training-free receptive field enlargement," in *Proc. ICLR*, 2025.
- [35] B. Lenz, O. Lieber, A. Arazi, A. Bergman, A. Manevich, B. Peleg, B. Aviram, C. Almagor, C. Fridman, D. Padnos, D. Gissin, D. Jannai, D. Muhlgay, D. Zimberg, E. M. Gerber, E. Dolev, E. Krakovsky, E. Safahi, E. Schwartz, G. Cohen, G. Shachaf, H. Rozenblum, H. Bata, I. Blass, I. Magar, I. Dalmedigos, J. Osin, J. Fadlon, M. Rozman, M. Danos, M. Gokhman, M. Zusman, N. Gidron, N. Ratner, N. Gat, N. Rozen, O. Fried, O. Leshno, O. Antverg, O. Abend, O. Dagan, O. Cohavi, R. Alon, R. Belson, R. Cohen, R. Gilad, R. Glozman, S. Lev, S. Shalev-Shwartz, S. H. Meirrom, T. Delbari, T. Ness, T. Asida, T. B. Gal, T. Braude, U. Pumerantz, J. Cohen, Y. Belinkov, Y. Globerson, Y. P. Levy, and Y. Shoham, "Jamba: Hybrid transformer-mamba language models," in *Proc. ICLR*, 2025.
- [36] L. Ren, Y. Liu, Y. Lu, yelong shen, C. Liang, and W. Chen, "Samba: Simple hybrid state space models for efficient unlimited context language modeling," in *Proc. ICLR*, 2025.
- [37] X. Dong, Y. Fu, S. Diao, W. Byeon, Z. Chen, A. S. Mahabaleshwarar, S.-Y. Liu, M. Van Keirsbilck, M.-H. Chen, Y. Suhara *et al.*, "Hymba: A hybrid-head architecture for small language models," in *Proc. ICLR*, 2025.
- [38] Zephyra Team, "Beta release of zonos-v0.1," <https://www.zephyra.com/post/beta-release-of-zonos-v0-1>, February 2025. [Online]. Available: <https://www.zephyra.com/post/beta-release-of-zonos-v0-1>
- [39] A. Bick, K. Li, E. P. Xing, J. Z. Kolter, and A. Gu, "Transformers to SSMs: Distilling quadratic knowledge to subquadratic models," in *NeurIPS*, 2024.
- [40] J. Wang, D. Paliotta, A. May, A. Rush, and T. Dao, "The mamba in the llama: Distilling and accelerating hybrid models," in *NeurIPS*, 2024.
- [41] A. Bick, T. Katsch, N. S. Sohoni, A. D. Desai, and A. Gu, "Llamba: Scaling distilled recurrent models for efficient language processing," in *First Workshop on Scalable Optimization for Efficient and Adaptive Foundation Models*, 2025.
- [42] G. E. Blelloch, "Prefix sums and their applications," 1990.
- [43] J. T. Smith, A. Warrington, and S. Linderman, "Simplified state space layers for sequence modeling," in *Proc. ICLR*, 2023.
- [44] J. Ko, S. Kim, T. Chen, and S.-Y. Yun, "Distillm: Towards streamlined distillation for large language models," in *Proc. ICML*, 2024.
- [45] H. Zen, V. Dang, R. Clark, Y. Zhang, R. J. Weiss, Y. Jia, Z. Chen, and Y. Wu, "Libritts: A corpus derived from librispeech for text-to-speech," in *Proc. Interspeech*, 2019.
- [46] P. Anastassiou, J. Chen, J. Chen, Y. Chen, Z. Chen, Z. Chen, J. Cong, L. Deng, C. Ding, L. Gao *et al.*, "Seed-tts: A family of high-quality versatile speech generation models," *arXiv preprint arXiv:2406.02430*, 2024.
- [47] A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, H. Lin, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Lin, K. Dang, K. Lu, K. Bao, K. Yang, L. Yu, M. Li, M. Xue, P. Zhang, Q. Zhu, R. Men, R. Lin, T. Li, T. Tang, T. Xia, X. Ren, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Wan, Y. Liu, Z. Cui, Z. Zhang, and Z. Qiu, "Qwen2.5 technical report," 2025.
- [48] Z. Ye, X. Zhu, C.-M. Chan, X. Wang, X. Tan, J. Lei, Y. Peng, H. Liu, Y. Jin, Z. Dai *et al.*, "Llms: Scaling train-time and inference-time compute for llama-based speech synthesis," *arXiv preprint arXiv:2502.04128*, 2025.
- [49] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, A. Yang, A. Fan, A. Goyal, A. Hartshorn, A. Yang *et al.*, "The llama 3 herd of models," 2024.
- [50] J. Shi, H.-j. Shim, J. Tian, S. Arora, H. Wu, D. Petermann, J. Q. Yip, Y. Zhang, Y. Tang, W. Zhang, D. S. Alharthi, Y. Huang, K. Saito, J. Han, Y. Zhao, C. Donahue, and S. Watanabe, "VERSA: A versatile evaluation toolkit for speech, audio, and music," in *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (System Demonstrations)*. Association for Computational Linguistics, 2025.