WAND: Windowed Attention and Knowledge Distillation for Efficient Autoregressive Text-to-Speech Models

Hanna Lee [1], *Tan Dat Nguyen* [1], *Jaehoon Kang* [2], *Kyuhong Shim* [2]

[1] Korea Advanced Institute of Science and Technology, Republic of Korea
[2] Sungkyunkwan University, Republic of Korea

hanna.lee@kaist.ac.kr, tandat.kaist@kaist.ac.kr, morateng@skku.edu, khshim@skku.edu

Abstract

Recent decoder-only autoregressive text-to-speech (AR-TTS) models produce high-fidelity speech, but their memory and compute costs scale quadratically with sequence length due to full self-attention. In this paper, we propose WAND, Windowed Attention and Knowledge Distillation, a framework that adapts pretrained AR-TTS models to operate with constant computational and memory complexity. WAND separates the attention mechanism into two: persistent global attention over conditioning tokens and local sliding-window attention over generated tokens. To stabilize fine-tuning, we employ a curriculum learning strategy that progressively tightens the attention window. We further utilize knowledge distillation from a full-attention teacher to recover high-fidelity synthesis quality with high data efficiency. Evaluated on three modern AR-TTS models, WAND preserves the original quality while achieving up to 66.2% KV cache memory reduction and length-invariant, near-constant per-step latency.

Index Terms: text-to-speech, speech synthesis, efficient TTS, global-local attention, WAND

1. Introduction

Text-to-Speech (TTS) models leveraging Transformer-based large language model (LLM) backbones have demonstrated the capability to generate near-human-quality speech from short reference audio [1, 2, 3, 4]. By utilizing neural audio codecs [5, 6], these models treat speech as discrete token sequences, enabling the direct adaptation of text-based LLMs for speech synthesis. Compared to conventional seq2seq TTS models such as Tacotron [7] and VITS [8], LLM-based TTS systems exhibit superior zero-shot speaker generalization, greater robustness to diverse linguistic inputs, and enhanced naturalness. However, the computational complexity of Transformer architectures scales quadratically with sequence length [9], leading to substantial memory overhead and high inference latency. This challenge hinders the deployment of LLM-based TTS models in stringent real-time applications [10].

To circumvent the scaling limitations of the Transformer, recent research has explored several avenues for reducing the computational intensity and memory requirements of LLM architectures. One prominent direction focuses on reducing the model depth through layer-wise pruning [11, 12], an approach recently applied to speech synthesis to minimize parameter counts [13]. However, depth reduction fails to address the fundamental nature of the self-attention mechanism, as the remaining layers still incur quadratic computational costs and identical per-layer memory overhead. On the other hand, architectural shifts toward linear-time mechanisms, such as Gated Linear Attention [14, 15] and Mamba-based architectures [16, 17, 18], offer significantly higher throughput. However, these paradigms necessitate training from scratch and often exhibit a performance gap, where the resulting speech quality and prosodic naturalness remain inferior to those achieved by established AR-TTS models. Other approaches accelerate inference by optimizing the decoding process, such as through speculative decoding [19, 20, 21] or parallel decoding [22, 23]. While this improves speed by generating multiple tokens per step, it still relies on self-attention and leaves the underlying memory scaling issues unresolved. A more common and simpler alternative is the use of Key-Value (KV) caching [24, 25], which is widely adopted in state-of-the-art models [3, 2]. While KV caching achieves a near-linear inference cost similar to linear attention while preserving the original model's performance, it introduces a critical bottleneck: the cumulative memory footprint continues to expand with every generated token [26, 27]. This restricts the synthesis of long-form utterances and imposes severe hardware constraints, a challenge that current LLM-based TTS systems have yet to resolve effectively.

In this work, we hypothesize that *AR-TTS models may not require full-sequence attention to maintain high-fidelity synthesis*. Specifically, we posit that the conditioning prompt tokens, comprising target text, reference audio, and auxiliary task or emotion tags, provide sufficient semantic and acoustic context for the generation. In contrast, generated speech tokens primarily serve to maintain local temporal consistency; thus, the model only requires a *localized view* of these tokens to track the decoding position relative to the monotonic speech trajectory. This is consistent with the "attention sink" phenomenon in text LLMs [28], where most of the attention mass is concentrated on several prefix tokens and a local window. Despite this, current AR-TTS architectures attend to all tokens indiscriminately, resulting in a linearly expanding KV cache that inherently limits long-sequence scalability.

Motivated by this view, we propose WAND, **W**indowed **A**ttention and K**n**owledge **D**istillation, to transform the scaling of computational and memory costs from linear to constant. Our framework bifurcates the attention mechanism into two distinct components: *Global Attention*, which maintains persistent access to the conditioning tokens (capturing 48–65% of attention mass), and *Local Sliding-Window Attention*, which restricts the receptive field for generated speech tokens to a fixed-size window. Unlike paradigms that necessitate training from scratch, WAND leverages pretrained weights, substantially reducing training cost and enabling seamless integration into existing AR-TTS pipelines. By employing knowledge distillation from a full-attention teacher, WAND recovers high-fidelity synthesis quality using only 53.8 hours of speech data [29].

We evaluate WAND across three different models: CosyVoice 2 [3], IndexTTS 1.5 [4], and SparkTTS [2]. For

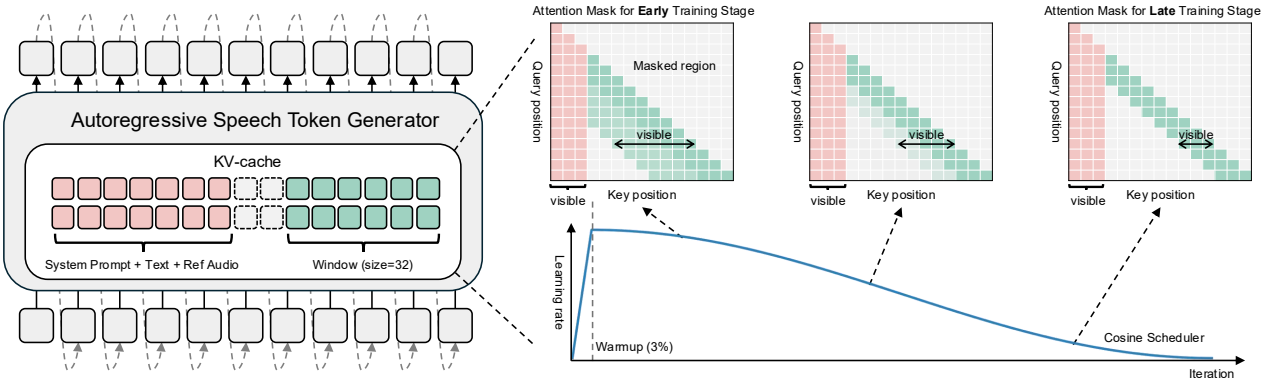

Figure 1: *Overview of the WAND framework. Conditioning tokens (system prompt, text, reference audio) retain global attention access, while generated acoustic tokens are restricted to a fixed-size sliding window of size W.*

10-second generation, WAND achieves up to a 66.2% reduction in KV cache size and significantly decreases cumulative computational costs, reducing total GFLOPs by as much as 46.9%. Notably, WAND achieves effective adaptation using only 53.8 hours of English data in a single epoch and generalizes to Mandarin without training data with a Character Error Rate (CER) degradation within 0.1% absolute. Our contributions are threefold: (1) an attention restriction method for LLM-based TTS ensuring constant memory and computational overhead without architectural modifications; (2) a data-efficient adaptation strategy via knowledge distillation that generalizes across languages; and (3) a cross-architecture validation achieving a bounded KV cache and constant per-step latency with negligible quality loss. We make our source code and demo publicly available. [1]

2. Method

2.1. Attention Restriction with Sliding Window Decoding

We posit that AR-TTS relies on two distinct types of information: *global context*, which determines the invariant characteristics of the generation, and *local context*, which governs fine-grained acoustic transitions [30, 13, 31]. By decoupling these, we can restrict the model's temporal attention to a local window without losing global consistency or speaker identity. Without loss of generality, we consider TTS conditioned on four fundamental inputs: a system prompt s that encodes global constraints, target text x, a reference audio prompt a^{pr} specifying speaker style, and the generated acoustic tokens $y_{1:T}$. These inputs are concatenated into a conditioning prompt:

$$p_\theta(y_{1:T} \mid s, x, a^{\mathrm{pr}}) = \prod_{t=1}^{T} p_\theta\left(y_t \mid y_{<t}, s, x, a^{\mathrm{pr}}\right). \quad (1)$$

As shown in Figure 1, we divide the attention mechanism into two distinct components. *Global Attention* (red blocks in Figure 1) provides fixed conditioning on s, x, and a^{pr}. Conversely, *Local Sliding-Window Attention* (green blocks in Figure 1) provides dynamic conditioning on the recent acoustic history $y_{<t}$. Because acoustic signals are locally coherent and monotonic, the influence of distant past tokens $y_{1:t-W-1}$ diminishes once global conditions are fixed. By restricting local attention to a window W, we not only maintain efficiency but also mitigate the propagation of sampling artifacts from the distant past:

$$p_\theta(y_t \mid y_{<t}, s, x, a^{\mathrm{pr}}) \approx p_\theta\left(y_t \mid y_{t-W:t-1}, s, x, a^{\mathrm{pr}}\right). \quad (2)$$

By partitioning the KV cache into a fixed global component and a rolling window of size W for acoustic tokens, we achieve a constant $\mathcal{O}(1)$ inference memory footprint relative to the total sequence length T. At the inference stage, this constant-length windowed attention enables arbitrarily long audio generation with bounded memory usage.

2.2. Knowledge Distillation for Attention Adaptation

While our hypothesis suggests that global context maintains core identity, some performance degradation is inevitable when a model trained on full attention is suddenly restricted to a local window. Our initial experiments show that it results in a slight decrease in content consistency. To mitigate this observed performance gap, we employ a knowledge distillation framework that supervises the restricted student model using two complementary objectives [29, 32, 33]. First, a cross-entropy loss ($\mathcal{L}_{\mathrm{CE}}$) anchors the student to the ground-truth acoustic tokens, ensuring fundamental alignment with the target speech. Second, a Skew Kullback–Leibler divergence [34] loss ($\mathcal{L}_{\mathrm{KL}}$) encourages the token probability distribution of the student to mimic the probability of a full-attention teacher model, both conditioned on the same historical context. This loss helps the student remain consistent when the long-range context is removed after attention adaptation. The final training objective is a weighted combination of both components: $\mathcal{L} = \mathcal{L}_{\mathrm{CE}} + \lambda\,\mathcal{L}_{\mathrm{KL}}$.

2.3. Curriculum Scheduling for Window Reduction

To stabilize fine-tuning, we introduce a curriculum that progressively reduces the window size from W_{start} to the target size W. Following a cosine schedule $\alpha(t)$ over T_c steps:

$$\alpha(t) = \frac{1}{2}\left(1 - \cos\left(\pi \min\left(\frac{t}{T_c}, 1\right)\right)\right), \quad (3)$$

the effective window size at step t is defined as $W(t) = W_{\text{start}} - \alpha(t)\,(W_{\text{start}} - W)$. To further ease adaptation, we apply a temperature-controlled soft mask to the attention logits A. Rather than immediate hard truncation, we define $A' =$

[1] https://onemeee.github.io/wand-tts/

Table 1: *Comparison of baselines and WAND models on* `test-en`*. Inference efficiency metrics (KV cache and GFLOPs) correspond to the accumulation required to generate 10 seconds of audio, with the KV cache reported in fp32 mode. All WAND models are fine-tuned solely on 53.8 hours of English data. NMOS is reported with a confidence interval of* 95%*.*

Model	Attn. Mechanism	Cache	Compute	KV Cache ↓	Reduction ↑	GFLOPs ↓	Speedup ↑	UTMOS ↑	NMOS ↑	SSIM ↑	WER ↓
Groundtruth	–	–	–	–	–	–	–	3.53	3.79 ± 0.18	100	1.44
CosyVoice2-0.5B	GQA ($W = \infty$)	$\mathcal{O}(L)$	$\mathcal{O}(L)$	10.48 MB	–	11.55	–	4.18	4.01 ± 0.17	**94.6**	1.94
	WAND ($W = 32$)	$\mathcal{O}(1)$	$\mathcal{O}(1)$	**5.25** MB	**49.9**%	**7.44**	**1.55**×	$\mathbf{4.21}_{+0.03}$	4.02 ± 0.16	$94.5_{-0.1}$	$\mathbf{1.72}_{-0.22}$
IndexTTS 1.5	MHA ($W = \infty$)	$\mathcal{O}(L)$	$\mathcal{O}(L)$	38.44 MB	–	6.18	–	3.97	4.13 ± 0.14	**94.4**	0.98
	WAND ($W = 32$)	$\mathcal{O}(1)$	$\mathcal{O}(1)$	**13.01** MB	**66.2**%	**3.28**	**1.89**×	$\mathbf{3.98}_{+0.01}$	4.13 ± 0.13	$\mathbf{94.4}_{+0.0}$	$\mathbf{0.91}_{-0.07}$
SparkTTS-0.5B	GQA ($W = \infty$)	$\mathcal{O}(L)$	$\mathcal{O}(L)$	18.09 MB	–	48.12	–	**3.93**	3.94 ± 0.16	92.6	3.27
	WAND ($W = 64$)	$\mathcal{O}(1)$	$\mathcal{O}(1)$	**7.15** MB	**60.5**%	**31.74**	**1.51**×	$\mathbf{3.93}_{+0.00}$	3.97 ± 0.16	$\mathbf{92.8}_{+0.2}$	$\mathbf{3.11}_{-0.16}$

Table 2: *Attention distribution of vanilla (full-attention) AR-TTS models during decoding, averaged over 5 long utterances.*

Model	Attention (%)		Local-W /Gen. (%)	Cov.
	Prompt	Generated		
CosyVoice2 ($W = 32$)	58.5	41.5	70.2	87.6
IndexTTS 1.5 ($W = 32$)	64.6	35.4	57.1	84.8
SparkTTS ($W = 64$)	47.9	52.1	82.8	91.0

$A - \tau(t)M$, where M is the binary mask for positions outside the current window. The masking strength $\tau(t)$ follows a log-scale interpolation:

$$\tau(t) = \exp\Big(\log \tau_{\text{start}} + \alpha(t)\,(\log \tau_{\text{end}} - \log \tau_{\text{start}})\Big). \quad (4)$$

As depicted in Figure 1, an initially small $\tau(t)$ permits partial attention to masked positions, preserving gradient flow during early optimization. As $\tau(t)$ increases, the model gradually adapts to strict inference-time constraints.

3. Experiments

Datasets & Fine-Tuning Configurations. We conduct fine-tuning experiments on the `train-clean-100` subset of LibriTTS [35], which contains approximately 53.8 hours of transcribed speech. Notably, we intentionally use only this modest subset rather than the full 585.80-hour corpus, demonstrating that WAND requires minimal data for effective adaptation. All models are fine-tuned using the AdamW optimizer with a cosine learning rate schedule and a peak learning rate of 1×10^{-5}. We fine-tune each baseline with WAND for 1 epoch on a single NVIDIA A100 MIG partition with 20 GB of memory.

Baselines. We evaluate WAND on three pretrained AR-TTS models spanning different architectures and codec configurations: **CosyVoice 2-0.5B** [3]: A 0.5B-parameter model built on the Qwen2.5-0.5B backbone, using an FSQ-based single-codebook codec with a token rate of 25 Hz; **IndexTTS 1.5** [4]: A GPT-style AR-TTS model employing a single-codebook VQ codec with a token rate of 25 Hz; and **SparkTTS-0.5B** [2]: A 0.5B-parameter model built on the Qwen2.5-0.5B [36] backbone, utilizing BiCodec with a semantic token rate of 50 Hz. This diversity in backbone architectures, codec designs (FSQ, VQ, BiCodec), and token rates (25 Hz vs. 50 Hz) provides a rigorous testbed for evaluating the generality of WAND.

Evaluations. We evaluate WAND using the Seed-TTS-eval [37] benchmark, performing tests on both `test-en` and `test-zh` to assess generation quality on both English and Mandarin.[2] For objective metrics, we calculate Word Error Rate (WER) using Whisper-large-v3 [38] and Character Error Rate (CER) using Paraformer-zh [39] as ASR systems to evaluate content accuracy for English and Mandarin, respectively. We also measure the speaker similarity (SSIM) as the cosine similarity between speaker embeddings extracted from the reference prompt and the synthesized speech using WavLM-based [40] x-vectors, while overall speech quality is assessed via the UTMOS metric [41]. Subjectively, we conduct Naturalness Mean Opinion Score (NMOS) tests on a 5-point scale involving 10 raters, with 50 randomly generated audio samples from each model to provide a measure of perceived naturalness.

4. Analysis

4.1. Target-Language Speech Quality

As demonstrated in Table 1, WAND maintains high-fidelity synthesis across all evaluated architectures on the Seed-TTS `test-en` benchmark. Notably, the absolute word error rate (WER) either remains within 0.2% of the baseline or exhibits slight improvement, as observed in CosyVoice 2, where the WER decreased from 1.94% to 1.72%. We attribute this performance stability—and occasional improvement— to the inherent regularization effect of the sliding window. By isolating the decoding context, WAND effectively mitigates the propagation of distant sampling artifacts and hallucinations. Consequently, this limits the model's dependency on its own prior outputs, mitigating the compounding errors typical of autoregressive generation, and forces the student model to rely more robustly on the invariant global conditioning.

4.2. Inference Efficiency

Beyond accuracy, WAND delivers substantial efficiency gains. For 10-second generation, memory usage is reduced by up to 66.2%, with IndexTTS 1.5 KV cache ($fp32$) decreasing from 38.44MB to 13.01MB. Computational cost is similarly reduced: CosyVoice 2 GFLOPs drop from 11.55 to 7.44, and IndexTTS from 6.18 to 3.28, yielding speedups of $1.51\times$–$1.89\times$. CosyVoice 2 and SparkTTS employ Grouped Query Attention [42]. FLOPs are computed using the full number of query heads, as KV heads are broadcast during attention. SparkTTS exhibits higher absolute FLOPs due to its higher token rate, resulting in double decoding steps compared to other baselines. The theoretical memory reduction translates directly to inference efficiency (Fig. 2). While full attention latency increases from 7.8ms to over 9.0ms with sequence length, WAND main-

[2] `https://github.com/BytedanceSpeech/seed-tts-eval`

Table 3: *Comparison of baselines and their fine-tuned version with WAND on* `test-zh`*. All WAND models are fine-tuned exclusively on English data.*

Model	Attn. Mechanism	UTMOS ↑	SSIM ↑	CER ↓
CosyVoice 2-0.5B	Baseline ($W = \infty$)	3.49	95.3	1.59
	WAND ($W = 32$)	3.50	95.3	1.53
IndexTTS 1.5	Baseline ($W = \infty$)	3.23	95.6	0.82
	WAND ($W = 32$)	3.25	95.6	0.91
SparkTTS-0.5B	Baseline ($W = \infty$)	3.28	94.2	2.14
	WAND ($W = 64$)	3.26	94.4	2.35

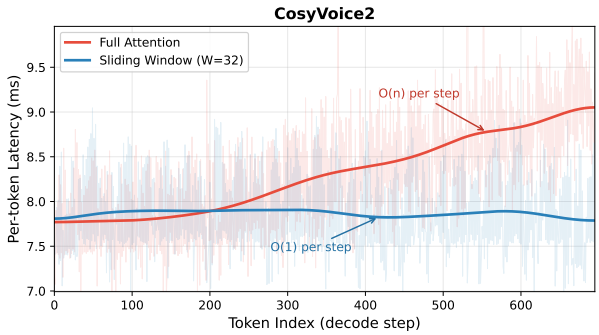

Figure 2: *Per-step decoding latency regarding the number of generated tokens. Full attention latency grows linearly with sequence length, while sliding-window attention maintains constant latency regardless of output length.*

tains near-constant latency due to bounded attention. Combined with bounded KV cache growth, this enables long-form synthesis with constant memory and constant per-step computation.

4.3. Data Efficiency and Cross-Lingual Preservation

Our framework achieves effective adaptation with only 53.8 hours of data and a single epoch of fine-tuning. Crucially, despite this English-only fine-tuning stage, WAND preserves the ability to generate high-quality Mandarin. As shown in Table 3, the performance on Chinese speech remains robust, with character error rate (CER) degradation limited to within 0.1% absolute for primary systems like CosyVoice 2 and IndexTTS 1.5. This cross-lingual resilience confirms that the learned attention restriction captures a universal structural property of speech (i.e., temporal locality) rather than language-specific patterns. Furthermore, WAND generalizes across three architecturally diverse systems—differing in backbone (Qwen2.5-based vs. GPT-style), codec design (FSQ, VQ, BiCodec), and token rate (25 Hz, 50 Hz)—without any architecture-specific modifications. This broad applicability confirms that the attention restriction is a universal optimization applicable to the AR-TTS paradigm, enabling consistent efficiency gains regardless of the underlying codec or model configuration.

4.4. Attention Pattern Analysis

To validate the design of our attention restriction, we analyze the attention patterns of vanilla full-attention models (Table 2). Across all three architectures, 48–65% of attention targets the conditioning prefix, confirming that these tokens serve as persistent anchors throughout generation and must retain global access. Within the decode region, 57–83% of attention concentrates on the most recent W tokens. Combining these components, the prefix and a local window jointly account for 85–91% of total attention, validating that restricting decode-region attention to a fixed window discards only a small fraction of the model's attention mass. Notably, SparkTTS allocates a larger share to the decode region (52%) and requires a proportionally wider window (W=64) to achieve comparable coverage, consistent with its higher codec token rate (50 Hz).

Table 4: *Ablation on distillation loss components. WER (%) on Seed-TTS-eval* `test-en`*.*

Loss	CosyVoice 2 W=32	IndexTTS 1.5 W=32	SparkTTS W=64
No distillation (SW only)	3.40	1.11	3.81
$\mathcal{L}_{\text{CE}}$ only	2.02	1.16	4.92
$\mathcal{L}_{\text{KL}}$ only	2.37	1.05	3.83
$\mathcal{L}_{\text{CE}} + \mathcal{L}_{\text{KL}}$	**1.72**	**0.91**	**3.11**

Table 5: *Ablation on curriculum learning strategy. WER (%) on Seed-TTS-eval* `test-en`*. Curriculum progressively reduces window size from 128 to W with temperature-scaled masking.*

Training Strategy	CosyVoice 2 W=32	IndexTTS 1.5 W=32
Direct ($W = 32$ from start)	1.86	1.03
Curriculum ($W = 128 \rightarrow 32$)	1.72	0.91

4.5. Ablation Studies

Distillation Loss Components. Table 4 isolates the contribution of each loss component. Applying the sliding window without distillation achieves a reasonable WER of 1.11% for IndexTTS 1.5, but substantially degrades performance for CosyVoice 2 (3.40%) and SparkTTS (3.81%). While adding either $\mathcal{L}_{\text{CE}}$ or $\mathcal{L}_{\text{KL}}$ alone recovers a portion of the performance gap for some models, only their combination consistently yields the best results across all three models. The result confirms that cross-entropy and KL divergence losses provide complementary supervision during attention adaptation.

Curriculum Learning. Table 5 compares direct training with the target window size against our curriculum strategy. The curriculum strategy begins with a window size of 128 and progressively narrows to the target W using temperature-scaled masking to gradually restrict attention over distant tokens. This approach consistently outperforms direct training, reducing WER from 1.86% to 1.72% for CosyVoice 2 and from 1.03% to 0.91% for IndexTTS 1.5.

5. Conclusion

In conclusion, **WAND** successfully transforms the computational and memory scaling of AR-TTS from linear to constant by bifurcating the attention mechanism into persistent *Global Attention* and *Local Sliding-Window Attention*. Our evaluations across CosyVoice 2, IndexTTS 1.5, and SparkTTS demonstrate that bounded KV caching enables constant per-step latency with negligible quality loss. Specifically, the framework achieved up to a 66.2% reduction in KV cache size and a significant decrease in GFLOPs, while maintaining robust cross-lingual generalization even with limited data. By overcoming the memory scaling limits inherent in Transformers, WAND enables long-form TTS with constant memory and per-step computation, paving the way for continuous, infinitely long audio generation without hardware constraints or performance degradation.

6. Acknowledgments

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2019-II190421, Artificial Intelligence Graduate School Program (Sungkyunkwan University)).

7. Generative AI Use Disclosure

Generative AI tools (Claude, Anthropic) were used to assist with grammar correction and polishing of the manuscript. All scientific content, experimental design, and analysis were conducted solely by the authors. All authors take full responsibility for the content of this paper.

8. References

[1] C. Wang, S. Chen, Y. Wu, Z. Zhang, L. Zhou, S. Liu, Z. Chen, Y. Liu, H. Wang, J. Li, L. He, S. Zhao, and F. Wei, "Neural codec language models are zero-shot text to speech synthesizers," *arXiv preprint arXiv:2301.02111*, 2023.

[2] X. Wang, M. Jiang, Z. Ma, Z. Zhang, S. Liu, L. Li, Z. Liang, Q. Zheng, R. Wang, X. Feng *et al.*, "Spark-tts: An efficient llm-based text-to-speech model with single-stream decoupled speech tokens," *arXiv preprint arXiv:2503.01710*, 2025.

[3] Z. Du *et al.*, "Cosyvoice 2: Scalable streaming speech synthesis with large language models," *arXiv preprint arXiv:2412.10117*, 2024.

[4] W. Deng, S. Zhou, J. Shu, J. Wang, and L. Wang, "Indextts: An industrial-level controllable and efficient zero-shot text-to-speech system," *arXiv preprint arXiv:2502.05512*, 2025.

[5] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," *Transactions on Machine Learning Research*, 2023.

[6] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, "Soundstream: An end-to-end neural audio codec," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 495–507, 2022.

[7] 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," in *Proc. Interspeech 2017*, 2017, pp. 4006–4010.

[8] J. Kim, J. Kong, and J. Son, "Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech," in *International conference on machine learning*. PMLR, 2021, pp. 5530–5540.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[10] 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 *Proceedings of the 42nd International Conference on Machine Learning*, vol. 267, 2025, pp. 48 245–48 261.

[11] A. Gromov, K. Tirumala, H. Shapourian, P. Glorioso, and D. Roberts, "The unreasonable ineffectiveness of the deeper layers," in *The Thirteenth International Conference on Learning Representations*, 2025.

[12] X. Men, M. Xu, Q. Zhang, Q. Yuan, B. Wang, H. Lin, Y. Lu, X. Han, and W. Chen, "ShortGPT: Layers in large language models are more redundant than you expect," in *Findings of the Association for Computational Linguistics: ACL 2025*, Jul. 2025, pp. 20 192–20 204.

[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 *2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2026, pp. 16 697–16 701.

[14] S. Yang, B. Wang, Y. Shen, R. Panda, and Y. Kim, "Gated linear attention transformers with hardware-efficient training," in *Proceedings of the 41st International Conference on Machine Learning*, ser. ICML'24. JMLR.org, 2024.

[15] T. Lemerle, T. Guichoux, A. Roebel, and N. Obin, "Lina-speech: Gated linear attention and initial-state tuning for multi-sample prompting text-to-speech synthesis," in *Proceedings of the AAAI 2026 Workshop on Audio-Centric AI: Towards Real-World Multimodal Reasoning and Application Use Cases (Audio-AAAI)*, ser. Proceedings of Machine Learning Research, vol. 312. PMLR, 2026, pp. 1–20.

[16] T. Dao and A. Gu, "Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality," in *International Conference on Machine Learning (ICML)*, 2024.

[17] 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 *2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025, pp. 1–5.

[18] T. D. Nguyen, S. Bae, J. S. Chung, and J.-H. Kim, "Mamtra: A hybrid mamba-transformer backbone for speech synthesis," *arXiv preprint arXiv:2603.12342*, 2026.

[19] Y. Leviathan, M. Kalman, and Y. Matias, "Fast inference from transformers via speculative decoding," in *International Conference on Machine Learning*. PMLR, 2023, pp. 19 274–19 286.

[20] 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 *2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025, pp. 1–5.

[21] B. Li, H. Wang, S. Zhang, Y. Guo, and K. Yu, "Fast and high-quality auto-regressive speech synthesis via speculative decoding," in *2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025.

[22] A. Santilli, S. Severino, E. Postolache, V. Maiorca, M. Mancusi, R. Marin, and E. Rodolà, "Accelerating transformer inference for translation via parallel decoding," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023, pp. 12 336–12 355.

[23] S. Kou, L. Hu, Z. He, Z. Deng, and H. Zhang, "Cllms: Consistency large language models," in *Forty-first International Conference on Machine Learning*, 2024.

[24] R. Pope, S. Douglas, A. Chowdhery, J. Devlin, J. Bradbury, A. Levskaya, J. Heek, K. Xiao, S. Agrawal, and J. Dean, "Efficiently scaling transformer inference," *Proceedings of machine learning and systems*, vol. 5, pp. 606–624, 2023.

[25] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. E. Gonzalez, H. Zhang, and I. Stoica, "Efficient memory management for large language model serving with pagedattention," in *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

[26] M. Kim, K. Shim, J. Choi, and S. Chang, "Infinipot: Infinite context processing on memory-constrained llms," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024, pp. 16 046–16 060.

[27] X. Liu, R. Li, Z. Liu, Q. Guo, Y. Song, K. Lv, H. Yan, L. Li, Q. Liu, and X. Qiu, "ReAttention: Training-free infinite context with finite attention scope," in *International Conference on Learning Representations*, 2025.

[28] G. Xiao, Y. Tian, B. Chen, S. Han, and M. Lewis, "Efficient streaming language models with attention sinks," in *The Twelfth International Conference on Learning Representations*, 2024.

[29] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[30] H. Yoon, C. Kim, E. Song, H.-W. Yoon, and H.-G. Kang, "Pruning Self-Attention for Zero-Shot Multi-Speaker Text-to-Speech," in *Interspeech 2023*, 2023, pp. 4299–4303.

[31] J. Kang, Y. Lee, Y. Park, and K. Shim, "Unlocking fine-grained and within-utterance speaking style control in prompt-based text-to-speech models," *arXiv preprint arXiv:2605.27376*, 2026.

[32] H. Inaguma and T. Kawahara, "Alignment knowledge distillation for online streaming attention-based speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 1371–1385, 2023.

[33] J. W. Yoon, "Heuristic-free knowledge distillation for streaming asr via multi-modal training," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 24, 2025, pp. 25 733–25 741.

[34] J. Ko, S. Kim, T. Chen, and S.-Y. Yun, "DistiLLM: Towards streamlined distillation for large language models," in *Proceedings of the 41st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 235, 2024, pp. 24 872–24 895.

[35] 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 *Interspeech 2019*, 2019, pp. 1526–1530.

[36] A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei *et al.*, "Qwen2.5 technical report," *arXiv preprint arXiv:2412.15115*, 2024.

[37] 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.

[38] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *International conference on machine learning*. PMLR, 2023, pp. 28 492–28 518.

[39] Z. Gao, Z. Li, J. Wang, H. Luo, X. Shi, M. Chen, Y. Li, L. Zuo, Z. Du, and S. Zhang, "FunASR: A Fundamental End-to-End Speech Recognition Toolkit," in *Interspeech 2023*, 2023, pp. 1593–1597.

[40] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao *et al.*, "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.

[41] T. Saeki, D. Xin, W. Nakata, T. Koriyama, S. Takamichi, and H. Saruwatari, "UTMOS: UTokyo-SaruLab System for VoiceMOS Challenge 2022," in *Interspeech 2022*, 2022, pp. 4521–4525.

[42] J. Ainslie, J. Lee-Thorp, M. de Jong, Y. Zemlyanskiy, F. Lebron, and S. Sanghai, "GQA: Training generalized multi-query transformer models from multi-head checkpoints," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 4895–4901.