SpeakStream: Streaming Text-to-Speech with Interleaved Data

Richard He Bai [1] **Tatiana Likhomanenko** [1] **Zijin Gu** [1] **Navdeep Jaitly** [1]

Abstract

The latency bottleneck of traditional text-to-speech (TTS) systems fundamentally hinders the potential of streaming large language models (LLMs) in conversational AI. These TTS systems, typically trained and inferenced on complete utterances, introduce unacceptable delays – even with optimized inference speeds – when coupled with streaming LLM outputs. This is particularly problematic for creating responsive conversational agents where low first-token latency is critical. In this paper, we present SpeakStream, a streaming TTS system that generates audio incrementally from streaming text using a decoder-only architecture. SpeakStream is trained using a next-step prediction loss on interleaved text-speech data. During inference, it generates speech incrementally while absorbing streaming input text, making it particularly suitable for cascaded conversational AI agents where an LLM streams text to a TTS system. Our experiments demonstrate that SpeakStream achieves state-of-the-art latency results in terms of first-token latency while maintaining the quality of non-streaming TTS systems. Our demo website is available at `https://apple.github.io/speakstream-demo/`.

1. Introduction

Speech interfaces for large language models (LLMs) have attracted growing interest (Borsos et al., 2023; Défossez et al., 2024; Xu et al., 2025). While end-to-end models that directly generate tokenized audio have been explored, cascaded systems—streaming text from an LLM to a TTS—consistently outperform end-to-end ones (Nguyen et al., 2025; Sakshi et al., 2024). The dominant cascaded-system bottleneck is latency, which has two sources: (1) waiting for the LLM to produce a complete text segment, and (2) waiting for the TTS to render audio. Recent work (Dang et al., 2024a; Dekel et al., 2024) addresses (1) with partial text windows, but these struggle with long-range dependencies and text-speech alignment (Dang et al., 2024a; Yang et al., 2024).

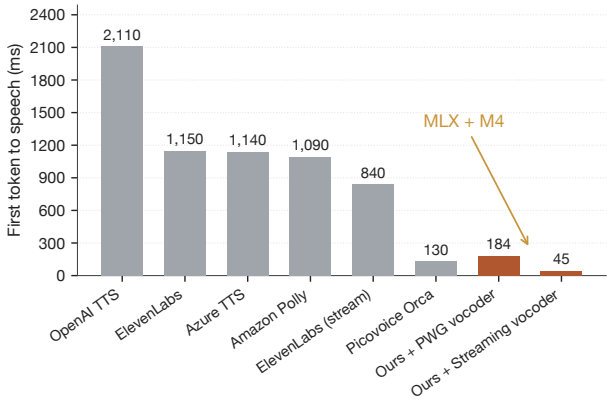

Figure 1. First-token to speech latency comparison. The 184ms (using ParallelWaveGAN (Yamamoto et al., 2020)) and 45ms (using a streaming vocoder) numbers are tested with MLX (Hannun et al., 2023) on a 64GB M4 Pro Mac Mini. Baseline latencies are taken from tts-latency-benchmark.

Source (2) is more fundamental: traditional TTS systems (Bai et al., 2024; 2022; Casanova et al., 2022; Du et al., 2024b; Gao et al., 2023; Ren et al., 2020; Wang et al., 2017; OpenAI, 2024) process complete text segments before emitting any audio. Inference accelerators such as StreamSpeech (Shopov et al., 2023) reduce the per-utterance compute, but cannot remove the inherent wait for the full text. This is especially problematic for interactive use, where the *first-token latency* (time from first input text token to first output audio token) must be minimized.

We propose **SpeakStream**, a decoder-only TTS that streams audio from streaming text by modeling *interleaved* text-speech segments. We construct interleaved data using a force-aligner (Bai et al., 2022) on standard TTS pairs, train the model with next-token prediction (loss only on speech tokens, with a beginning-of-speech and end-of-speech marker around each speech segment), and at inference emit a speech segment after each new text segment of length m, caching all prior text and speech in the kv-cache. This unified decoder-only design eliminates explicit alignment at inference and makes streaming a natural mode of operation rather than a

[1]Apple. Correspondence to: Richard He Bai <richard-bai@apple.com>.

Published at ICML 2026 Workshop on Machine Learning for Audio.

retrofit.

Empirically, SpeakStream achieves the lowest WER across all latency configurations, and human evaluators rate its coherence comparable to non-streamed RichTTS (Bai et al., 2024). Deployed on a Mac mini (M4 Pro, 64GB, 2024), it reaches 30ms TTS latency (Figure 1), making it suitable for real-time interactive applications.

2. Related Work

Traditional TTS systems (Wang et al., 2017; Casanova et al., 2022; Ren et al., 2020; Gao et al., 2023; Du et al., 2024b) process complete text to generate complete audio, and most encoder-based architectures (FastSpeech2's transformer encoder, Tacotron's LSTM, E3 TTS's BERT (Koroteev, 2021)) must re-encode their input as new text arrives, struggling to synthesize natural speech with limited context. Autoregressive speech generation, as in RichTTS (Bai et al., 2024) and VALL-T (Du et al., 2024a), offers smoother chunk transitions but does not by itself solve dual streaming.

Prior dual-streaming attempts have notable limitations. Dekel et al. (2024) distill a non-streaming TTS into a non-attentive Tacotron (Shen et al., 2020), but show limited zero-shot capability. Dang et al. (2024a) extend LiveSpeech (Dang et al., 2024b) to text-chunk synthesis, but suffer alignment issues that require an auxiliary CTC-ASR aligner, complicating the pipeline. Transducer-based TTS (Kim et al., 2023) offers cleaner alignment but is underexplored for dual streaming.

Concurrently, Yang et al. (2024) also use a decoder-only model for dual streaming, but interleave text and speech at a fixed ratio rather than from alignment. Fixed ratios poorly handle speaking-rate variation and break the precise text-audio correspondence needed for interruptible agents. Our approach uses alignment-driven interleaving and supports both Scheme 1 (with text repetition) and Scheme 2 (without).

3. Method

Our approach enables streaming TTS through an interleaved representation of text and speech tokens in a decoder-only architecture. This section describes the model, interleaving schemes, and inference.

Token Representation. We use character-level embeddings for text: each word w_t consists of x character embeddings $c_{t_1}, \cdots, c_{t_x}$. For speech, we adopt dMel (Bai et al., 2024) which discretizes mel-filterbank channels into intensity bins, yielding for each word an audio chunk s_t of y dMel embeddings $f_{t_1}, \cdots, f_{t_y}$.

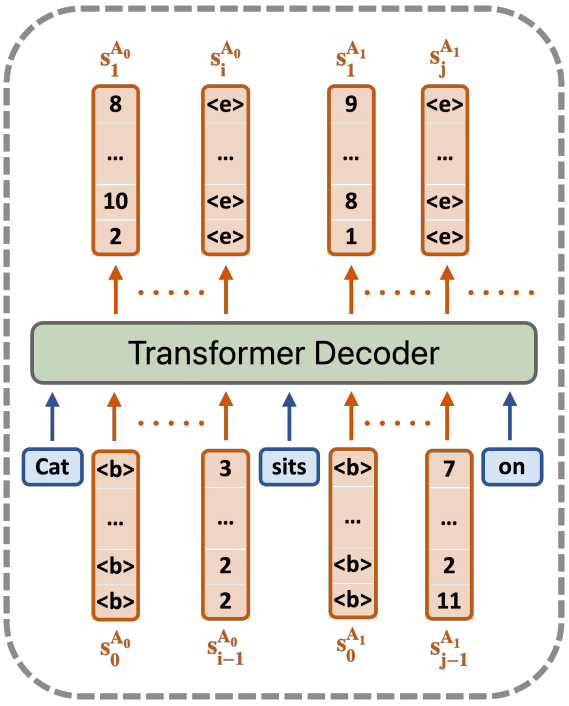

Figure 2. SpeakStream transformer decoder architecture.

3.1. SpeakStream Model

SpeakStream is built upon a vanilla transformer decoder similar to RichTTS (Bai et al., 2024). While RichTTS feeds the model the concatenated sequence $[w_{\text{bos}}, w_1, \cdots, w_t, w_{\text{eos}}, s_{\text{bos}}, s_1, \cdots, s_t, s_{\text{eos}}]$, SpeakStream interleaves text and speech as $[T_1, A_1, \cdots, T_x, A_x]$, where each T_i is a chunk of text tokens and each $A_i = (s_{\text{bos}}, s_i, s_{\text{eos}})$ is a chunk of speech tokens. To establish precise temporal correspondence between words and speech frames, we utilize the A3T's alignment mechanism (Bai et al., 2022).

By training a decoder-only transformer on interleaved sequences, SpeakStream learns to synthesize A_i conditioned on T_i, $A_{<i}$, and $T_{<i}$. Compared to existing streaming solutions that synthesize each text chunk independently, this design (1) preserves complete speech and text history via kv-cache, ensuring acoustic coherence and semantic precision; (2) supports high-quality synthesis even with short text segments; and (3) eliminates the need for an explicit force aligner at inference, since the model predicts the EOS token at each chunk boundary.

3.2. Interleaving Schemes

Streaming TTS faces a latency-accuracy trade-off: shorter T_i reduces latency, but heteronyms and prosody require lookahead context. We address this by parameterizing the **text window length** m and **speech hop length** n ($1 \leq n \leq m$): the first n words of T_i correspond to A_i, while the

remaining $(m-n)$ words supply future context.

We study two schemes. **Scheme 1** repeats text tokens across chunks:

$$T_i = w_{n(i-1)+1}, \cdots, w_{\min(t, n(i-1)+m)}$$
$$A_i = s_{\text{bos}}, s_{n(i-1)+1}, \cdots, s_{\min(t, n\cdot i)}, s_{\text{eos}}$$

Example (m=3, n=2, t=8):

$$[w_1, w_2, w_3, s_{\text{bos}}, s_1, s_2, s_{\text{eos}}, w_3, w_4, w_5, s_{\text{bos}}, s_3, s_4, s_{\text{eos}},$$
$$w_5, w_6, w_7, s_{\text{bos}}, s_5, s_6, s_{\text{eos}}, w_7, w_8, s_{\text{bos}}, s_7, s_8, s_{\text{eos}}]$$

Scheme 2 avoids text repetition by emitting each new text token only once, at the cost of more complex attention patterns since A_i's corresponding words become separated by variable-length gaps determined by prior speech durations.

3.3. Streaming Inference

At inference, SpeakStream proceeds autoregressively: it (1) accumulates the first m words and generates the corresponding dMel tokens, converted into waveforms in tandem by a streaming Mel-to-wave vocoder; (2) absorbs each subsequent text segment and continues generation by updating the kv-cache; (3) repeats until the input stream ends. The primary latency comes from accumulating the first m words, making m a direct knob for the latency-quality trade-off.

4. Experiments

Setup. We use the LJSpeech (Ito & Johnson, 2017) dataset (single-speaker, 22kHz). Following RichTTS (Bai et al., 2024), our model has 36 transformer decoder layers (258M parameters); the dMel feature uses a 25ms hop, 80 channels, 16 bins, with a ParallelWaveGAN vocoder (Yamamoto et al., 2020). We train each model for 100k steps using Adam (lr=1e-3, 5k warmup, cosine schedule, gradient clipping 1.0) with a batch of 1 hour on A100s in BF16. The baselines are RichTTS (Bai et al., 2024) and XTTS (Gölge & The Coqui TTS Team, 2021), the best open-source streaming-capable TTS. We score generated speech with WhisperX (*base.en*) (Bain et al., 2023; Radford et al., 2023) WER. For the end-to-end latency analysis, we additionally train a simple streaming vocoder on LibriTTS-R (Koizumi et al., 2023) that emits waveforms at single-frame latency, used solely as a tool for measuring the system-level latency upper bound.

Main Results. Table 1 shows that directly applying RichTTS or XTTS to streaming segments severely degrades quality (WER >68% and >222% respectively for unigram synthesis). XTTS in particular hallucinates phonemes, leading to high insertion errors. In contrast, SpeakStream achieves ∼7% WER for unigram synthesis and below 5% with more context. **SpeakStream's performance at m=5, n=1 matches RichTTS's full-text synthesis (3.38 vs. 3.28 WER) while using only 5 words of latency instead of the entire input.**

Scheme 1 consistently outperforms Scheme 2 because S1 places the words corresponding to A_i adjacent to it, while S2's variable-length gaps (induced by prior speech durations) create more complex attention patterns. We adopt S1 going forward. Configurations with $m = n$ underperform, indicating extra context words help; performance also degrades for $m > 5$ due to excessive text repetition. We hence use m=5, n=1 as the default.

Human Evaluation. We sampled 100 segments from the LJSpeech dev set and collected 7 ratings each (284 unique annotators) for naturalness and coherence. We compare 4-gram and 6-gram RichTTS/XTTS streaming variants against (m=4, n=2) and (m=6, n=2) SpeakStream (Table 2). RichTTS and XTTS degrade severely under streaming, especially in coherence, while SpeakStream's coherence matches non-streaming RichTTS, indicating it preserves prosodic continuity across segments.

Latency Analysis. We measure SpeakStream's end-to-end latency on Mac Mini 2024 (M4 Pro, 64GB) using MLX (Hannun et al., 2023), evaluating 25 sentences from LibriSpeech (Panayotov et al., 2015) dev-clean with n=1 S1 configurations. We decompose latency into three components: *TTS latency* (first text token to first Mel frame), *vocoder latency* (first Mel frame to first audio sample), and *total latency*. To distinguish first audio output from first *spoken phoneme*, we prompt SpeakStream with a brief silence (200–400ms) and verify the first generated Mel frame contains speech rather than silence—models that emit initial silence can otherwise game first-byte latency.[1]

Table 3 reports the breakdown. SpeakStream's TTS latency is ∼30ms across all configurations, since it only requires accumulating m words before generation. The vocoder is the dominant cost: non-streaming PWG must accumulate 10 Mel frames (∼150ms) before producing audio. Pairing SpeakStream with a single-frame streaming vocoder reduces vocoder latency to 13ms—an order-of-magnitude drop—and brings the full system below 50ms total, a 2.4× improvement over the 120ms threshold considered favorable for interactive applications (Défossez et al., 2024), demonstrating SpeakStream's suitability for real-time conversational AI.

5. Conclusion

We presented SpeakStream, a decoder-only streaming TTS system that synthesizes speech from streaming text through

[1]The tts-latency-benchmark measures first-byte latency without this filter.

Table 1. WER of SpeakStream with Scheme 1 (S1) and Scheme 2 (S2) evaluated by WhisperX ASR (base.en). The WER of groundtruth audio is 2.09.

	Hop n=1		Hop n=2		Hop n=3		Hop n=4		Hop n=5		Hop n=6		∞
XTTS-V2	222.28		45.01		30.92		28.97		17.13		15.28		**3.67**
RichTTS	68.18		30.20		20.30		16.81		13.07		10.55		**3.28**
SpeakStream	S1	S2	S1	S2	S1	S2	S1	S2	S1	S2	S1	S2	
Window m=1	7.47		-		-		-		-		-		-
Window m=2	**4.50**	5.26	7.18		-		-		-		-		-
Window m=3	**3.99**	6.88	**4.19**	5.11	4.78		-		-		-		-
Window m=4	**3.88**	6.16	**3.80**	5.09	**4.36**	5.35	5.21		-		-		-
Window m=5	**3.38**	5.59	**3.61**	6.09	**3.65**	4.82	4.73	**4.59**	4.52		-		-
Window m=6	**4.26**	6.20	**3.61**	4.36	**3.93**	4.52	**4.36**	6.14	**4.86**	4.95	4.30		-

Table 2. Human evaluation results for synthesized speech naturalness and coherence (95% confidence interval).

	NonStreaming	Streaming (m=4)	Streaming (m=6)
		Naturalness	
GroundTruth	**4.4** $\pm$ 0.1	-	-
RichTTS	3.8 $\pm$ 0.1	2.2 $\pm$ 0.1	2.5 $\pm$ 0.1
XTTS	3.9 $\pm$ 0.1	2.1 $\pm$ 0.1	2.5 $\pm$ 0.1
SpeakStream	-	**3.7** $\pm$ 0.1	**3.5** $\pm$ 0.1
		Coherence	
GroundTruth	**4.2** $\pm$ 0.0	-	-
RichTTS	3.9 $\pm$ 0.1	2.3 $\pm$ 0.1	2.2 $\pm$ 0.1
XTTS	4.1 $\pm$ 0.1	1.8 $\pm$ 0.1	2.7 $\pm$ 0.1
SpeakStream	-	**3.9** $\pm$ 0.1	**3.8** $\pm$ 0.1

Table 3. Latency (ms) of SpeakStream with ParallelWaveGAN (PWG) or a streaming vocoder, tested on Apple Silicon (Mac mini, M4 Pro, 64GB, 2024). Player latency is less than 0.2ms.

SpeakStream	Vocoder	TTS (ms)	Vocoder (ms)	Total (ms)
Window m=3	PWG	$28_{\pm 2}$	$152_{\pm 12}$	$180_{\pm 14}$
Window m=4		$33_{\pm 8}$	$150_{\pm 5}$	$183_{\pm 13}$
Window m=5		$34_{\pm 9}$	$150_{\pm 6}$	$184_{\pm 15}$
Window m=3	Streaming	$27_{\pm 2}$	$13_{\pm 1}$	$40_{\pm 3}$
Window m=4		$28_{\pm 2}$	$13_{\pm 1}$	$41_{\pm 4}$
Window m=5		$29_{\pm 1}$	$13_{\pm 1}$	$45_{\pm 15}$

interleaved text-speech modeling. SpeakStream closes the quality gap with non-streaming TTS, achieving WER comparable to full-text synthesis while operating with $\sim$30ms first-token TTS latency on a Mac Mini (M4 Pro, 64GB). Human evaluations confirm that coherence is preserved across streaming segments. Future work includes extending the approach to multi-speaker, larger-scale, and cross-lingual settings.

References

Bai, H., Zheng, R., Chen, J., Ma, M., Li, X., and Huang, L. A^{3}T: Alignment-aware acoustic and text pretraining for speech synthesis and editing. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 1399–1411. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/bai22d.html.

Bai, H., Likhomanenko, T., Zhang, R., Gu, Z., Aldeneh, Z., and Jaitly, N. dmel: Speech tokenization made simple. *arXiv preprint arXiv:2407.15835*, 2024.

Bain, M., Huh, J., Han, T., and Zisserman, A. Whisperx: Time-accurate speech transcription of long-form audio. *INTERSPEECH 2023*, 2023.

Borsos, Z., Marinier, R., Vincent, D., Kharitonov, E., Pietquin, O., Sharifi, M., Roblek, D., Teboul, O., Grangier, D., Tagliasacchi, M., et al. Audiolm: a language modeling approach to audio generation. *IEEE/ACM transactions on audio, speech, and language processing*, 31: 2523–2533, 2023.

Casanova, E., Weber, J., Shulby, C. D., Junior, A. C., Gölge, E., and Ponti, M. A. Yourtts: Towards zero-shot multi-speaker tts and zero-shot voice conversion for everyone. In *International Conference on Machine Learning*, pp. 2709–2720. PMLR, 2022.

Dang, T., Aponte, D., Tran, D., Chen, T., and Koishida, K. Zero-shot text-to-speech from continuous text streams. *arXiv preprint arXiv:2410.00767*, 2024a.

Dang, T., Aponte, D., Tran, D., and Koishida, K. Livespeech: Low-latency zero-shot text-to-speech via autoregressive modeling of audio discrete codes. *arXiv preprint arXiv:2406.02897*, 2024b.

Défossez, A., Mazaré, L., Orsini, M., Royer, A., Pérez, P., Jégou, H., Grave, E., and Zeghidour, N. Moshi: a speech-text foundation model for real-time dialogue. *arXiv preprint arXiv:2410.00037*, 2024.

Dekel, A., Shechtman, S., Fernandez, R., Haws, D., Kons, Z., and Hoory, R. Speak while you think: Streaming speech synthesis during text generation. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 11931–11935. IEEE, 2024.

Du, C., Guo, Y., Wang, H., Yang, Y., Niu, Z., Wang, S., Zhang, H., Chen, X., and Yu, K. Vall-t: Decoder-only generative transducer for robust and decoding-controllable text-to-speech. *arXiv preprint arXiv:2401.14321*, 2024a.

Du, Z., Chen, Q., Zhang, S., Hu, K., Lu, H., Yang, Y., Hu, H., Zheng, S., Gu, Y., Ma, Z., et al. Cosyvoice: A scalable multilingual zero-shot text-to-speech synthesizer based on supervised semantic tokens. *arXiv preprint arXiv:2407.05407*, 2024b.

Défossez, A., Mazaré, L., Orsini, M., Royer, A., Pérez, P., Jégou, H., Grave, E., and Zeghidour, N. Moshi: a speech-text foundation model for real-time dialogue, 2024. URL https://arxiv.org/abs/2410.00037.

Gao, Y., Morioka, N., Zhang, Y., and Chen, N. E3 tts: Easy end-to-end diffusion-based text to speech. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 1–8. IEEE, 2023.

Gölge, E. and The Coqui TTS Team. Coqui TTS: A deep learning toolkit for Text-to-Speech, battle-tested in research and production, 1 2021. URL https://www.coqui.ai.

Hannun, A., Digani, J., Katharopoulos, A., and Collobert, R. MLX: Efficient and flexible machine learning on apple silicon, 2023. URL https://github.com/ml-explore.

Ito, K. and Johnson, L. The lj speech dataset. https://keithito.com/LJ-Speech-Dataset/, 2017.

Kim, M., Jeong, M., Choi, B. J., Lee, D., and Kim, N. S. Transduce and speak: Neural transducer for text-to-speech with semantic token prediction. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 1–7. IEEE, 2023.

Koizumi, Y., Zen, H., Karita, S., Ding, Y., Yatabe, K., Morioka, N., Bacchiani, M., Zhang, Y., Han, W., and Bapna, A. Libritts-r: A restored multi-speaker text-to-speech corpus. In *Proc. Interspeech 2023*, pp. 5496–5500, 2023.

Koroteev, M. V. Bert: a review of applications in natural language processing and understanding. *arXiv preprint arXiv:2103.11943*, 2021.

Nguyen, T. A., Muller, B., Yu, B., Costa-Jussa, M. R., Elbayad, M., Popuri, S., Ropers, C., Duquenne, P.-A., Algayres, R., Mavlyutov, R., et al. Spirit-lm: Interleaved spoken and written language model. *Transactions of the Association for Computational Linguistics*, 13:30–52, 2025.

OpenAI. Text-to-speech guide, 2024. URL https://platform.openai.com/docs/guides/text-to-speech.

Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 5206–5210. IEEE, 2015.

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., and Sutskever, I. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*, pp. 28492–28518. PMLR, 2023.

Ren, Y., Hu, C., Tan, X., Qin, T., Zhao, S., Zhao, Z., and Liu, T.-Y. Fastspeech 2: Fast and high-quality end-to-end text to speech. *arXiv preprint arXiv:2006.04558*, 2020.

Sakshi, S., Tyagi, U., Kumar, S., Seth, A., Selvakumar, R., Nieto, O., Duraiswami, R., Ghosh, S., and Manocha, D. Mmau: A massive multi-task audio understanding and reasoning benchmark. *arXiv preprint arXiv:2410.19168*, 2024.

Shen, J., Jia, Y., Chrzanowski, M., Zhang, Y., Elias, I., Zen, H., and Wu, Y. Non-attentive tacotron: Robust and controllable neural tts synthesis including unsupervised duration modeling. *arXiv preprint arXiv:2010.04301*, 2020.

Shopov, G., Gerdjikov, S., and Mihov, S. Streamspeech: Low-latency neural architecture for high-quality on-device speech synthesis. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2023. doi: 10.1109/ICASSP49357.2023.10096566.

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

Xu, J., Guo, Z., He, J., Hu, H., He, T., Bai, S., Chen, K., Wang, J., Fan, Y., Dang, K., Zhang, B., Wang, X., Chu, Y., and Lin, J. Qwen2.5-omni technical report. *arXiv preprint arXiv:2503.20215*, 2025.

Yamamoto, R., Song, E., and Kim, J.-M. Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6199–6203. IEEE, 2020.

Yang, Y., Ma, Z., Liu, S., Li, J., Wang, H., Meng, L., Sun, H., Liang, Y., Xu, R., Hu, Y., et al. Interleaved speech-text language models are simple streaming text to speech synthesizers. *arXiv preprint arXiv:2412.16102*, 2024.