Chatterbox-Flash: Prior-Calibrated Block Diffusion for Streaming Zero-Shot TTS

Anonymous ACL submission

Abstract

We present Chatterbox-Flash, a zero-shot text-to-speech model obtained by fine-tuning a pretrained autoregressive TTS decoder into a block-diffusion decoder, enabling parallel token generation within each block while retaining block-by-block streaming. We find that naively transferring mainstream block-diffusion decoding to discrete speech tokens degrades quality, as a long-tail token distribution biases parallel position selection toward a few high-frequency tokens. To mitigate this without architectural modification, we introduce two inference-time techniques: prior-calibrated scoring, which subtracts the block-level marginal token distribution, and an early-decoding schedule, which adaptively terminates iteration based on calibrated confidence. On standard zero-shot TTS benchmarks, Chatterbox-Flash attains high-fidelity synthesis comparable to strong autoregressive and non-autoregressive baselines, while supporting streaming inference with time-to-first-packet on par with streaming AR systems and substantially lower real-time factor.

1 Introduction

Zero-shot text-to-speech (TTS), capable of synthesizing speech in unseen speakers' voices from a short reference audio clip, has emerged as a central capability of modern speech synthesis, driven by large-scale multi-speaker training (Wang et al., 2023; Le et al., 2023; Borsos et al., 2023; Anastassiou et al.).

Recent zero-shot TTS models can be broadly categorized along two axes. Along the axis of generation order, autoregressive (AR) models generate tokens sequentially, conditioning each step on previous outputs (Wang et al., 2023; Du et al., 2024, 2025; Zhou et al., 2026), while non-autoregressive (NAR) models produce all positions in parallel (Eskimez et al., 2024; Chen et al., 2025; Wang et al., 2025). Along the axis of modeling space, they operate either on discrete tokens derived from neural audio codecs (Défossez et al., 2022; Zeghidour et al., 2021) or on continuous latent representations (Shen et al., 2024). Among these combinations, AR language models over discrete audio codecs have proven particularly effective, delivering strong speech quality while remaining amenable to streaming inference (Yang et al., 2026).

Despite these advantages, AR decoding incurs an inherent latency since tokens are produced sequentially and wall-clock time grows linearly with output length—a bottleneck that cannot be removed by engineering alone, motivating model-native parallel decoding. Among such approaches, diffusion language models (DLMs) (Austin et al., 2021; Lou et al., 2024; Sahoo et al., 2024) generate multiple tokens per step and have recently achieved throughput far above AR models with minimal quality loss, with the LLaDA family (Nie et al., 2026; Bie et al., 2025, 2026) and other extensions (Li et al., 2026; Ye et al., 2025a) scaling the paradigm to discrete large language models (dLLMs). Block Diffusion (Arriola et al., 2025b) adds block-causal training and inference compatible with streaming, later made practical at scale by Fast-dLLM (Wu et al., 2025b) and Fast-dLLM v2 (Wu et al., 2025a).

However, DLMs remain underexplored for speech, with existing efforts limited in scale (Ku et al., 2026) or lacking native streaming (Zhu et al., 2026). Moreover, we find that DLM decoding techniques do not transfer directly to discrete speech: codec sequences are heavily skewed toward a few dominant tokens—notably silence—that carry little context-dependent information (Liu et al., 2025; Sicherman and Adi, 2023), and block-by-block decoding restricts position selection to a small local window (Shu et al., 2026) where ranking is fragile and not directly supervised (Asano et al., 2026).

In this work, we present Chatterbox-Flash, a zero-shot TTS model obtained by fine-tuning a

pretrained autoregressive decoder into a discrete block-diffusion decoder, retaining the original architecture and replacing only the training objective with masked denoising. To address the problem with degraded quality, we introduce two inference-time techniques—*prior-calibrated scoring* and an *early-decoding schedule*—and integrate them with a block-causal streaming pipeline that delivers streaming inference at substantially lower latency than AR baselines.

Our contributions are as follows:

- **Streaming Block-diffusion TTS** To our knowledge, the first zero-shot TTS model that combines block-diffusion decoding with native block-by-block streaming over discrete audio codecs.
- **Prior-Calibrated Scoring** An inference-time correction that suppresses the long-tail token bias in parallel position selection, requiring no architectural change or additional forward pass.
- **Early-Decoding Schedule** An adaptive termination rule that lowers the average number of denoising steps below the maximum budget based on calibrated confidence.
- **Empirical Validation** Chatterbox-Flash matches strong AR and NAR baselines in quality while being the only system in our comparison with native streaming support.

2 Method

2.1 Modeling

Architecture Our system extends Chatterbox-TTS (Resemble AI, 2025)[1], an open-source two-stage zero-shot TTS pipeline. Stage 1 is a Llama-style Transformer decoder (T3) that performs next-token prediction over a discrete speech token sequence $\mathbf{y} = (y_1, \ldots, y_T)$ extracted at 25 Hz by a neural audio codec. The decoder is conditioned on

$$\mathbf{c} = [\mathbf{e}_s, \mathbf{x}_{\text{text}}, \mathbf{x}_{\text{speech}}], \tag{1}$$

which combines a global speaker embedding $\mathbf{e}_s$ obtained from a GE2E-trained voice encoder (Wan et al., 2020), the input text token sequence $\mathbf{x}_{\text{text}}$, and the prompt speech tokens $\mathbf{x}_{\text{speech}}$ extracted from the reference audio. The speech distribution is factorized autoregressively as

$$p(\mathbf{y} \mid \mathbf{c}) = \prod_{t=1}^{T} p(y_t \mid y_{<t}, \mathbf{c}). \tag{2}$$

Stage 2 is a flow-matching vocoder that converts the generated tokens to waveforms with chunk-wise streaming.

Block Diffusion Following (Arriola et al., 2025b; Wu et al., 2025a), we apply discrete denoising diffusion block-by-block on the T3 decoder instead of over the full sequence. A length-T sequence x is partitioned into $B = \lceil T/D \rceil$ non-overlapping blocks $x^{(1)}, \ldots, x^{(B)}$ of size D, giving

$$p(x) = \prod_{b=1}^{B} p\left(x^{(b)} \mid x^{(<b)}\right). \tag{3}$$

Each factor is modeled by a parallel masked predictor: positions in $x^{(b)}$ are randomly replaced with $[\text{M}]$ to form $x_t^{(b)}$, and the predictor recovers the masked tokens in parallel from $x^{(<b)}$ and $x_t^{(b)}$. The left-to-right inter-block factorization naturally supports block-wise streaming generation.

2.2 Training

Packed Input and Hybrid Attention The packed input $[\mathbf{c}, x_t]$ uses a hybrid attention scheme over B speech blocks (Figure 3): causal over the conditioning $\mathbf{c}$, bidirectional within each speech block, and causal across blocks, so that $x_t^{(b)}$ attends only to $\mathbf{c}$ and $x^{(<b)}$. Unlike Fast-dLLM v2 (Wu et al., 2025a), which is block-causal throughout, we keep $\mathbf{c}$ causal to preserve the pretrained backbone's embedding space and apply block-diffusion attention only to the speech part, maintaining the monotonic text-to-speech alignment. We implement this with custom attention kernels (Appendix A).

Complementary Masking At each training step we sample $t \sim \mathcal{U}(\epsilon, 1-\epsilon)$, derive the per-token mask probability from a fixed noise schedule, and draw a binary mask $m \in \{0,1\}^T$ over the speech positions to form x_t ($x_{t,i} = y_i$ if $m_i = 0$, else $[\text{M}]$). Following Fast-dLLM v2 (Wu et al., 2025a), we add the complementary view $\bar{m} = 1 - m$ as a second sample in the same batch, so every position is supervised under both masked and unmasked contexts.

[1] https://github.com/resemble-ai/chatterbox

Token-Shift Denoising Loss We adopt a next-token prediction parameterization following Fast-dLLM v2 (Wu et al., 2025a): a masked position i is predicted from the hidden state at position $i-1$ rather than from the mask token itself. This shifted-label form preserves the backbone's autoregressive interface while still allowing bidirectional context within each block. The per-token cross-entropy loss at a masked position i is

$$\ell_i = -\log p_\theta\Big(y_i \mid \mathbf{c}, x^{(<b(i))}, x_t^{(b(i))}\Big), \quad (4)$$

where y_i is the clean target, $b(i)$ is the block containing i, and the conditioning $\mathbf{c}$, the clean preceding blocks $x^{(<b(i))}$, and the noised current block $x_t^{(b(i))}$. With $\mathcal{M}_b = \{i \mid x_{t,i} = [\mathrm{M}]\}$ the masked positions in block b (excluding padding), the sample-level loss averages per-token losses within each block and then over blocks,

$$\mathcal{L}_{\text{denoise}} = \frac{1}{B}\sum_{b=1}^{B}\frac{1}{|\mathcal{M}_b|}\sum_{i\in\mathcal{M}_b}\ell_i. \quad (5)$$

2.3 Inference

2.3.1 Block-Autoregressive Decoding

Block-diffusion inference commits the sequence one block at a time in left-to-right order, with masked positions in the current block unmasked in parallel. Already-committed blocks act as clean context, and their key-value caches are appended sequentially to provide the inter-block context for subsequent blocks. Since the conditioning prefix is encoded causally and never attends to speech tokens, its key-value cache depends only on $\mathbf{c}$ and is computed once at the start of inference and reused across every block. At each step, the forward pass therefore operates only on the current block, accessing the prefix and previously committed blocks through their appended caches.

At each step within a block, the forward pass produces a predictive distribution $p_i^{(k)}$ and a predicted token $\hat{x}_i^{(k)} = \arg\max_v p_i^{(k)}(v)$ at every masked position i. Two decisions must then be made: *which* positions to commit at this step (Section 2.3.2) and *how many* positions to commit (Section 2.3.3). When classifier-free guidance is used (Section 2.3.4), $p_i^{(k)}$ refers to the conditional branch alone, while a separate guidance combination determines $\hat{x}_i^{(k)}$.

2.3.2 Prior-Calibrated Scoring

Block-level commitment can induce *boundary-induced context truncation* (BICT) (Shu et al., 2026): once a few positions in a block are committed to incorrect tokens, subsequent blocks decode on top of corrupted context. This is particularly pronounced for discrete speech codecs, where strong local acoustic dependencies between adjacent frames coexist with a small set of *dominant tokens*—such as silence or low-energy frames—that occupy a disproportionate share of the marginal distribution (Liu et al., 2025; Asano et al., 2026); mistakenly committing such a token near a block boundary breaks acoustic continuity.

To address this, we propose *prior-calibrated scoring*, which assigns each masked position an ordering score for parallel unmasking. The common choice is the model's per-position confidence $p_i^{(k)}(\hat{x}_i^{(k)})$, but for discrete speech codecs this tends to assign large values to dominant tokens regardless of context, causing them to be unmasked preferentially. To separate this marginal bias from the contextual prediction, we use a pointwise mutual information (PMI) score,

$$s_i^{(k)} = \log p_i^{(k)}(\hat{x}_i^{(k)}) - \log \bar{p}(\hat{x}_i^{(k)}), \quad (6)$$

where $\bar{p}$ is a reference distribution measuring the marginal probability of producing $\hat{x}_i^{(k)}$ irrespective of local context. The first term is the model's log-confidence at position i, and the second subtracts the marginal share of the same token under $\bar{p}$, so that $s_i^{(k)}$ measures how specifically the predicted token is licensed by local context.

A natural choice for $\bar{p}$ is the in-block marginal averaged over the current block's predictive distributions, but this prior is itself shaped by $\mathbf{c}$, making the score partially self-referential. We instead use the *unconditional block prior*, computed once from a single forward pass on all-masked sequence $[\mathrm{M}]^D$ with conditioning removed,

$$\bar{p}(v) = \frac{1}{D}\sum_{j=1}^{D} p_\theta\big(v \,\big|\, [\mathrm{M}]^D, \mathbf{c} = \varnothing\big)_j. \quad (7)$$

Since $\bar{p}$ depends only on (D, θ), it is cached for the lifetime of the model.

2.3.3 Unmasking Schedule

Beyond which positions to unmask, the number of positions unmasked at each step must also be chosen. Committing too many positions in a single

step risks introducing incorrect tokens as misleading context, while committing too few requires nearly all K steps and increases inference cost.

Time-Shifted Schedule LaViDa (Li et al., 2026) introduced a time-shifted (TS) schedule that biases unmasking away from uniform, also adopted in OmniVoice (Zhu et al., 2026) as its decoding rule: at each step, the TS schedule determines *how many* positions to unmask, and the model commits the corresponding number of top-confidence positions. We build on the same TS schedule for time allocation, replacing only the position-selection criterion with prior-calibrated scoring (Section 2.3.2). The cumulative fraction of unmasked tokens at step k follows

$$r_k = \frac{\tau \cdot (k/K)}{1 + (\tau - 1) \cdot (k/K)}, \tag{8}$$

with $r_0 = 0$, total steps K, and shift parameter τ; the fraction of newly unmasked tokens at step k is $f_k = r_k - r_{k-1}$. The optimal τ differs across configurations such as K and the target token distribution, and we explored different τ values in our experiments.

Early Decoding We further adapt the unmasking fraction at each step based on the prior-calibrated scores $s_i^{(k)}$ of Equation (6). At step k, positions whose score exceeds a threshold θ_k are unmasked,

$$\theta_k = \mathrm{Quantile}\big(\{s_i^{(k)}\}_{i \in \mathcal{M}},\, q_k\big), \tag{9}$$

$$q_k = \max\big(0,\, 1 - \alpha \cdot \tfrac{k+1}{K}\big)\,, \tag{10}$$

where $\mathcal{M}$ is the set of masked positions in the current block and $\alpha \in [0, 1]$ controls how quickly the threshold relaxes. At early steps, q_k is close to one and only a few positions in the upper region are unmasked; as k increases, q_k decreases and more positions become eligible. Denoting the resulting fraction by $g_k = q_{k-1} - q_k$, we combine the two schedules via $\max(f_k, g_k)$, so that the TS schedule provides per-step commits while early decoding adds further commits when calibrated confidence is high. Decoding terminates when all positions in the block have been unmasked or when K steps have been reached; under large α, the average step count drops below K. Following OmniVoice (Zhu et al., 2026), we additionally support sampling temperatures at both the token and position levels, the latter realized as Gumbel-perturbed selection over the prior-calibrated scores with temperature $\beta = 5$; their effect is studied in Section 3.5.

2.3.4 Classifier-Free Guidance

We combine classifier-free guidance (CFG) (Ho and Salimans, 2022) with prior-calibrated decoding by running both a conditional and an unconditional forward at each step, the latter with $\mathbf{c}$ zeroed, yielding logits ℓ_i^c and ℓ_i^u. Token sampling uses the standard combination $\ell_i = (1 + w)\ell_i^c - w\ell_i^u$ to determine the predicted token $\hat{x}_i^{(k)} = \arg\max_v \mathrm{softmax}(\ell_i)_v$, while the prior-calibrated score is evaluated on the conditional branch alone (i.e., $p_i^{(k)} = \mathrm{softmax}(\ell_i^c)$ in Equation (6)). This decoupling—CFG-guided sampling for the committed token, conditional-only PMI for position ranking—keeps the ranking insensitive to w while still routing the CFG-guided token through Equation (6). We use $w = 1.0$ by default; a full sweep of w and an alternative mode combining s_i^c and s_i^u are provided in Appendix C.2.

3 Experiments

3.1 Training Data

We train on approximately 70k hours of English speech (44M utterances, 528k speakers), compiled from publicly available corpora—spanning large-scale read speech (Pratap et al., 2020; He et al., 2024; Parcollet et al., 2025; Koizumi et al., 2023; Bakhturina et al., 2021), expressive and anechoic speech (Nguyen et al., 2023; Richter et al., 2024), and accented English (Wang et al., 2024)—together with privately collected audiobook, conversational, and short-form utterances (names, numbers, time expressions), detailed in Appendix B.

3.2 Experimental Setup

Evaluation Benchmarks We evaluate on two publicly available English zero-shot TTS benchmarks. LibriSpeech-PC test-clean is a zero-shot voice cloning benchmark built on the test-clean split of LibriSpeech-PC (Meister et al., 2023), and Seed-TTS test-en is the English evaluation set introduced by Seed-TTS (Anastassiou et al.).

Metrics We report SIM-o (cosine similarity between WavLM-ECAPA-TDNN (Desplanques et al., 2020; Chen et al., 2022) speaker embeddings of generated vs. reference speech), WER (HuBERT (Hsu et al., 2021) ASR transcription on LibriSpeech-PC, Whisper-large-v3 (Radford et al., 2022) on Seed-TTS test-en, against the input text), and UTMOS (Saeki et al., 2022) for naturalness.

Model	#Params	Steps	LibriSpeech-PC test-clean			Seed-TTS test-en		
			SIM-o↑	WER↓	UTMOS↑	SIM-o↑	WER↓	UTMOS↑
Ground-truth	–	–	0.690	1.87	4.10	0.734	2.14	3.52
Autoregressive Models								
IndexTTS2 (Zhou et al., 2026)	1.7B	–	0.700	2.35	4.06	0.706	2.33	3.65
CosyVoice3 (Du et al., 2025)	1.1B	–	0.694	**1.59**	4.28	0.696	2.17	3.96
VoxCPM (Zhou et al., 2025)	0.7B	–	**0.717**	1.74	4.18	**0.731**	1.92	3.77
Qwen3-TTS (Hu et al., 2026)	1.1B	–	0.704	1.60	**4.41**	0.708	**1.54**	**4.16**
Chatterbox (Resemble AI, 2025)	0.5B	–	0.707	1.99	4.29	0.685	2.20	4.10
Non-Autoregressive Models								
F5-TTS (Chen et al., 2025)	0.4B	–	0.655	1.89	3.89	0.664	1.85	3.72
ZipVoice (Zhu et al., 2025)	0.1B	–	0.668	1.64	3.98	0.697	1.70	3.82
MaskGCT (Wang et al., 2025)	2.2B	–	0.691	2.26	3.91	0.713	2.88	3.55
OmniVoice-Emilia (Zhu et al., 2026)	0.8B	–	0.697	1.57	4.23	0.717	1.72	3.88
OmniVoice (Zhu et al., 2026)	0.8B	–	**0.729**	**1.30**	4.28	**0.741**	**1.60**	3.91
Block-Autoregressive Models (Chatterbox-Flash, Ours)								
w/ Fast-dLLM v2 decoding†	0.5B	10	0.656	15.36	4.14	0.646	14.49	4.00
w/ TS schedule	0.5B	8	0.714	1.69	**4.29**	0.703	1.97	**4.09**
w/ **PMI** ($\alpha = 0$, ours)	0.5B	8	**0.717**	**1.67**	**4.29**	**0.704**	**1.96**	**4.09**
w/ **PMI + ED** ($\alpha = 0.5$, ours)	0.5B	**6.4**	0.713	1.67	4.28	0.704	2.04	4.08

Table 1: Objective evaluation on zero-shot TTS benchmarks. Best results within the AR / NAR / Block-AR groups are in **bold**. Baseline numbers are taken from OmniVoice (Zhu et al., 2026). **Steps**: average denoising steps per block for Chatterbox-Flash (lower is better; not applicable to AR/NAR baselines). The Block-AR variants share the same canonical configuration ($D = 16$, $\tau = 0.5$, $w = 1.0$, $T = 0.2$, $\beta = 5$), differing only in the decoding method: Fast-dLLM v2's threshold-based decoding (at $K = 10$), TS schedule (top-confidence selection, as in OmniVoice; $K = 8$), our prior-calibrated scoring (PMI, $K = 8$), and PMI with early decoding (PMI+ED, $\alpha = 0.5$, $K = 8$ with early termination). †Fast-dLLM v2 uses threshold 0.3, top-p 0.95, small batch size of 16.

Metric configuration and baseline numbers are taken from OmniVoice (Zhu et al., 2026).

Implementation Details The model replaces the T3 decoder of Chatterbox-TTS with the block-diffusion architecture described in Section 2, trained with the hybrid attention mask and the token-shift loss. We initialize from a pretrained Chatterbox-TTS checkpoint and continue training with AdamW using a cosine learning rate schedule (peak 10^{-5}, 10% warmup) with an effective batch size of 440 in bfloat16 precision. The model is trained with a block size of $D = 32$. Text inputs are normalized by a custom preprocessor that converts numbers, dates, times, and other non-standard tokens into their spoken forms before tokenization. At inference time, our canonical configuration is block size $D = 16$, number of denoising steps $K = 8$, TS schedule parameter $\tau = 0.5$, CFG scale $w = 1.0$, sampling temperature 0.2, and position temperature $\beta = 5$. We report two main settings differing only in the early-decoding parameter α: a quality-strongest setting at $\alpha = 0$ (no early decoding) and an efficiency-oriented setting at $\alpha = 0.5$. All experiments are conducted on NVIDIA H100 GPUs, and inference uses attention kernels and paged key-value cache management from FlashInfer (Ye et al., 2025b); further implementation details are provided in Appendix E.

3.3 Main Results

Table 1 reports zero-shot TTS performance against recent state-of-the-art autoregressive (AR) and non-autoregressive (NAR) baselines. We report the strongest objective numbers under the canonical configuration without early decoding ($\alpha = 0$, $K = 8$); the quality–compute trade-off enabled by early decoding is reported separately in Table 2.

Comparison with Chatterbox A comparison with Chatterbox, the AR backbone we build on, isolates the effect of the AR-to-block-diffusion conversion. On LibriSpeech-PC, Chatterbox-Flash improves SIM-o (0.717 vs. 0.707) and WER (1.67 vs. 1.99), and matches UTMOS (4.29 vs. 4.29). On Seed-TTS test-en, SIM-o and WER improve (0.704 vs. 0.685, 1.96 vs. 2.20) while UTMOS is essentially unchanged (4.09 vs. 4.10). Our model

Benchmark	WER↓		Steps/blk	
	α=0	α=0.5	α=0	α=0.5
LibriSpeech-PC	**1.67**	1.67	8	**6.47**
Seed-TTS test-en	**1.96**	2.04	8	**6.10**

Table 2: Step-budget comparison between Chatterbox-Flash without ($\alpha = 0$) and with ($\alpha = 0.5$) early decoding at $K = 8$ (quality metrics in Table 1). Early decoding reduces the average step count while keeping WER within noise of the no-ED baseline, exploiting quality saturation at $K = 8$ to convert excess steps into compute savings.

unlocks parallel decoding with preserved or improved perceptual quality.

Comparison with NAR Baselines Within the NAR group, Chatterbox-Flash is second on LibriSpeech-PC SIM-o (0.717, trailing OmniVoice at 0.729) and second on LibriSpeech-PC WER (1.67, behind OmniVoice at 1.30), despite training on 70k hours of English versus OmniVoice's 581k hours of multilingual data. It achieves the best UTMOS among them on LibriSpeech-PC (4.29) and is competitive on Seed-TTS (4.09), surpassing F5-TTS, MaskGCT, and OmniVoice-Emilia on WER as well. Chatterbox-Flash is also the only NAR system in this comparison that natively supports streaming inference (Appendix G).

Effect of Decoding and Early Decoding The *Block-Autoregressive Models* group in Table 1 compares four decoding methods under the same canonical configuration. Fast-dLLM v2 decoding transfers poorly to discrete speech codecs (WER > 14 on both benchmarks), despite the carefully tuned threshold from our observation. The TS schedule alone (top-confidence selection, as in OmniVoice) and our prior-calibrated scoring (PMI, $\alpha = 0$) achieve statistically comparable quality, with SIM-o and UTMOS indistinguishable. PMI's decisive contribution is therefore not raw quality but the calibration of its scores, which serves as a reliable thresholding signal for early decoding: PMI+ED ($\alpha = 0.5$) reduces the average step count from 8 to 6.47 on LibriSpeech-PC and to 6.10 on Seed-TTS at no WER cost / $+0.08$ WER respectively (Table 2), with SIM-o and UTMOS unchanged. PMI+ED thus matches the TS schedule baseline's quality at $\sim$20% fewer steps.

Streaming Efficiency We further measure two streaming-oriented metrics against Qwen3-TTS (Hu et al., 2026), an autoregressive streaming TTS model with publicly reported numbers on comparable hardware: *time to first packet* (TTFP), the wall-clock time from receiving the request to emitting the first audio packet, and *real-time factor* (RTF), the ratio of generation wall-clock time to synthesized audio duration. Both metrics are computed on 50 utterances at concurrency 1, measured up to the moment the first audio packet is emitted by the server. Under the efficiency-oriented configuration ($D = 16$, $\alpha = 0.5$), Chatterbox-Flash attains TTFP 118 ms and RTF 0.107 (Table 3). TTFP is competitive with Qwen3-TTS, falling below all 25 Hz variants (138–150 ms) and within $\sim$20 ms of the 12 Hz variants (97–101 ms). The decisive gap appears in sustained throughput: Chatterbox-Flash's RTF of 0.107 is roughly 2.7× lower than the closest Qwen3-TTS variant (12 Hz, 0.6B at 0.288) and 2.4× lower than the Qwen3-TTS-25Hz-1.7B baseline (0.253), corresponding to $\sim$9× real-time synthesis even on a single concurrent request. Larger blocks combined with more aggressive early decoding push this further: at $D = 32$ and $\alpha = 0.75$, Chatterbox-Flash reaches TTFP 103 ms (matching the fastest Qwen3-TTS-12Hz variant) and RTF 0.076, synthesizing audio at $\sim$13× real time—roughly 3.8× the throughput of any Qwen3-TTS configuration.

Model / Config	TTFP (ms)↓	RTF↓
Qwen3-TTS (Hu et al., 2026) (autoregressive)		
25 Hz, 1.7B	150	0.253
25 Hz, 0.6B	138	0.234
12 Hz, 1.7B	101	0.313
12 Hz, 0.6B	97	0.288
Chatterbox-Flash (25 Hz, 0.5B, ours)		
$D = 16$, $\alpha = 0.5$ (default)	118	0.107
$D = 16$, $\alpha = 0.75$	106	0.091
$D = 24$, $\alpha = 0.5$	119	0.100
$D = 24$, $\alpha = 0.75$	105	0.084
$D = 32$, $\alpha = 0.5$	115	0.090
$D = 32$, $\alpha = 0.75$	**103**	**0.076**

Table 3: Streaming efficiency at concurrency 1, measured over 50 utterances. **TTFP**: wall-clock time from request to first audio packet emission. **RTF**: ratio of generation wall-clock time to synthesized audio duration. Best Chatterbox-Flash values are in **bold**. Qwen3-TTS numbers are taken from its technical report.

3.4 Human Evaluation

To complement the objective metrics, we conducted a side-by-side evaluation against ElevenLabs v3, a frontier commercial zero-shot TTS

Metric	ElevenLabs v3	Chatterbox-Flash
NMOS mean↑	**4.04**	3.91
$\% \leq 2 \downarrow$	12.9	**8.6**
$\% \geq 4 \uparrow$	**80.0**	67.1
SMOS mean↑	3.50	**4.56**

Table 4: Human evaluation on 10 utterances randomly sampled from Seed-TTS test-en, with 7 ratings per utterance (70 ratings per system in total). NMOS: naturalness; SMOS: speaker similarity to reference, both on 5-point Likert scales.

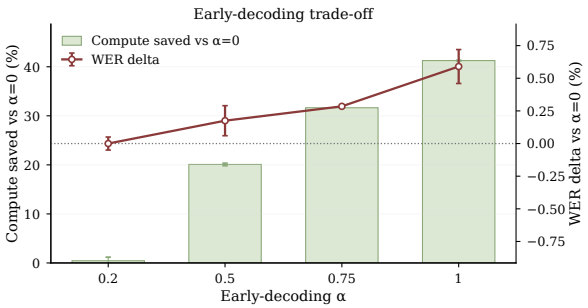

Figure 1: Early-decoding trade-off at $D = 16$, $K = 10$, averaged across LibriSpeech-PC and Seed-TTS test-en. Bars: number of computing steps saved relative to $\alpha = 0$. Line: WER delta versus $\alpha = 0$, with error bars spanning the two benchmarks. At $\alpha = 0.5$, $\sim$20% steps are saved at negligible WER cost; further savings up to $\sim$41% at $\alpha = 1$ incur a $\sim$+0.6 WER increase.

system. Listeners rated naturalness (NMOS) and speaker similarity to the reference audio (SMOS) on independent 5-point Likert scales, on 10 utterances randomly sampled from Seed-TTS test-en with 7 ratings per utterance, yielding 70 ratings per system. On NMOS (Table 4), Chatterbox-Flash and ElevenLabs v3 attain comparable means (3.91 vs. 4.04), but Chatterbox-Flash exhibits a cleaner low tail: no MOS-1 ratings and only 8.6% MOS-$\leq$ 2 ratings, compared to 4.3% and 12.9% for ElevenLabs v3. On SMOS, Chatterbox-Flash is rated substantially higher (4.56 vs. 3.50), indicating that it preserves reference-speaker characteristics more faithfully despite the smaller training corpus.

3.5 Ablation Studies

We ablate two inference-time hyperparameters affecting the quality–compute trade-off at the canonical configuration: the block size D and the denoising step budget K in combination with the early-decoding parameter α.

Block Size Figure 2(a) sweeps the inference block size $D \in \{8, 16, 24, 32\}$ with the model trained at $D = 32$. SIM-o and UTMOS are essentially flat across the range. WER stays within noise up to $D = 16$ (1.51 $\rightarrow$ 1.61 on LibriSpeech-PC, 1.93 $\rightarrow$ 1.75 on Seed-TTS) before degrading sharply at $D \geq 24$ (2.38 and 4.14 at $D = 24, 32$ on LibriSpeech-PC; 2.45 and 4.10 on Seed-TTS). The WER degradation at $D \geq 24$ reflects a difficulty parallel-unmasking on large blocks: these configurations require committing more positions per step than the model can confidently rank, even with prior calibration.

Step Budget Figure 2(b) characterizes the effect of step size K in fixed-step decoding at $D = 16$. $K \leq 2$ cannot recover the speech sequence (WER above 10, clipped from the display), but quality stabilizes rapidly: WER reaches 1.92 on LibriSpeech-PC and 2.16 on Seed-TTS at $K = 4$, and plateaus by $K \geq 6$. We adopt $K = 8$ as the default, providing modest headroom over the $K = 6$ plateau for adaptive early termination. The stars in Figure 2(b) mark adaptive early decoding at $\alpha = 1$, which reaches the same plateau-level WER at $\sim$4.6 average steps—a $\sim$41% reduction relative to the $K = 10$ sweep budget without exceeding the fixed-step plateau.

Early Decoding Figure 1 sweeps the early-decoding parameter $\alpha \in \{0.2, 0.5, 0.75, 1.0\}$ at $D = 16$, $K = 10$. At $\alpha = 0.2$ early decoding is almost inactive (<1% steps saved): per-step commits are dominated by the TS schedule. From $\alpha = 0.5$ onward the rule meaningfully tightens the step budget: 20% steps saved at $\alpha = 0.5$, 32% at $\alpha = 0.75$, and 41% at $\alpha = 1.0$. The WER cost grows mildly with α, remaining within +0.6 of the no-ED baseline even at $\alpha = 1.0$. We adopt $\alpha = 0.5$ as the canonical efficiency-oriented setting, providing $\sim$20% compute reduction at negligible quality cost; $\alpha = 0.75$–1.0 offer further savings for latency-critical deployments.

Further Ablations Sweeps of the CFG scale w, sampling temperature T, and position temperature β, together with a head-to-head comparison of the TS schedule baseline against PMI across step budgets $K \in \{2, 5, 8\}$, are provided in Appendix C. The comparison shows that PMI and the TS schedule baseline track each other closely across the step range, confirming that the choice of decoding method contributes only marginally to quality at any compute budget—PMI's advantage lies in pro-

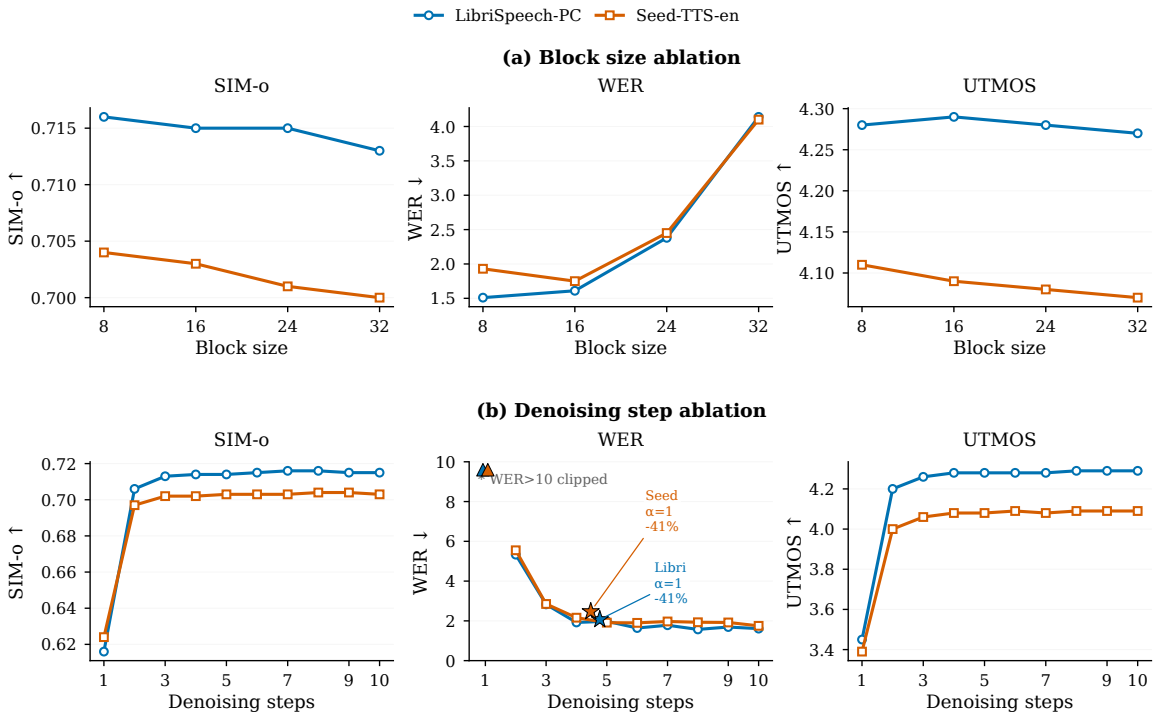

Figure 2: Inference-time ablations on LibriSpeech-PC (blue, circles) and Seed-TTS test-en (orange, squares). **(a) Block size** D: SIM-o and UTMOS are essentially flat across $D \in \{8, 16, 24, 32\}$; WER stays within noise up to $D = 16$ before degrading sharply at $D \geq 24$, while inference latency drops monotonically (Table 3). **(b) Denoising step budget** K: fixed-step decoding ($\alpha = 0$, $K \in \{1, \ldots, 10\}$) recovers speech only at $K \geq 3$ (WER for $K \leq 2$ exceeds the display range and is marked by triangles at the top of the WER panel); quality plateaus by $K \geq 6$. Stars mark adaptive early decoding at $\alpha = 1$ on each benchmark, attaining plateau-level quality at $\sim$4.6 average steps per block—a $\sim$41% reduction relative to the $K = 10$ budget—without exceeding the fixed-step plateau.

viding the calibrated confidence signal that enables adaptive early decoding.

4 Conclusion

We presented Chatterbox-Flash, a block-diffusion zero-shot TTS model obtained by fine-tuning a pretrained autoregressive decoder into a parallel masked decoder while preserving block-by-block streaming. Our key contributions are an inference-time prior-calibrated scoring scheme, which suppresses the dominant-token bias of discrete speech codecs and provides a well-calibrated confidence signal; an early-decoding schedule that adaptively terminates iteration; and a streaming-compatible inference engine that combines the two. On standard zero-shot TTS benchmarks, it matches strong AR and NAR baselines in quality despite training on substantially less data, achieves the highest UTMOS among NAR baselines on both LibriSpeech-PC and Seed-TTS, and reduces the average denoising step count by $\sim$20% at minimal quality cost. These results indicate that block diffusion with calibration-aware inference is a viable design point for production-grade streaming zero-shot TTS.

Limitations

We prioritized training stability under the fixed mixed-data setting of Section 3.1 over data-related ablations, leaving the contribution of individual data sources unisolated. The model also collapses when fine-tuned with substantially larger block sizes ($D \geq 128$) even with prior-calibrated decoding, while OmniVoice's full-sequence formulation remains stable; alternative recipes we explored (Appendix F) reach larger D under restricted conditions but introduce prosody collapse or sampling sensitivity, leaving the parallelism gap with full-sequence models open. Finally, at saturated compute our prior-calibrated scoring (PMI) and the TS schedule baseline yield statistically equivalent objective metrics (Table 7); PMI's principal advantage in our setup is therefore the calibrated confidence it provides for adaptive early decoding rather than a direct quality gain, and whether prior calibration helps under tighter budgets or out-of-domain references is left to future work.

References

Sand. ai, Hansi Teng, Hongyu Jia, Lei Sun, Lingzhi Li, Maolin Li, Mingqiu Tang, Shuai Han, Tianning Zhang, W. Q. Zhang, Weifeng Luo, Xiaoyang Kang, Yuchen Sun, Yue Cao, Yunpeng Huang, Yutong Lin, Yuxin Fang, Zewei Tao, Zheng Zhang, and 20 others. 2025. Magi-1: Autoregressive video generation at scale. *Preprint*, arXiv:2505.13211.

Philip Anastassiou, Jiawei Chen, Jitong Chen, Yuanzhe Chen, Zhuo Chen, Ziyi Chen, Jian Cong, Lelai Deng, Chuang Ding, Lu Gao, and 1 others. Seed-tts: A family of high-quality versatile speech generation models, 2024. *URL https://arxiv. org/abs/2406.02430*.

Marianne Arriola, Naveen Venkat, Jonathan Granskog, and Anastasis Germanidis. 2025a. Adapting autoregressive vision language models for parallel diffusion decoding. `https://runwayml.com/research/autoregressive-to-diffusion-vlms`. Runway Research.

Marianne Arriola and 1 others. 2025b. Block diffusion: Interpolating between autoregressive and diffusion language models. *arXiv preprint arXiv:2503.09573*.

Hikaru Asano, Tadashi Kozuno, Kuniaki Saito, and Yukino Baba. 2026. Where-to-unmask: Ground-truth-guided unmasking order learning for masked diffusion language models. *arXiv preprint arXiv:2602.09501*.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. 2021. Structured denoising diffusion models in discrete state-spaces. In *NeurIPS*.

Evelina Bakhturina, Vitaly Lavrukhin, Boris Ginsburg, and Yang Zhang. 2021. Hi-fi multi-speaker english tts dataset. *Preprint*, arXiv:2104.01497.

Tiwei Bie, Maosong Cao, Xiang Cao, Bingsen Chen, Fuyuan Chen, Kun Chen, Lun Du, Daozhuo Feng, Haibo Feng, Mingliang Gong, and 1 others. 2026. Llada2.1: Speeding up text diffusion via token editing. *arXiv preprint arXiv:2602.08676*.

Tiwei Bie, Maosong Cao, Kun Chen, Lun Du, Mingliang Gong, Zhuochen Gong, Yanmei Gu, Jiaqi Hu, Zenan Huang, Zhenzhong Lan, and 1 others. 2025. Llada2.0: Scaling up diffusion language models to 100b. *arXiv preprint arXiv:2512.15745*.

Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, and 1 others. 2023. Audiolm: a language modeling approach to audio generation. *IEEE/ACM transactions on audio, speech, and language processing*, 31:2523–2533.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, Jian Wu, Long Zhou, Shuo Ren, Yanmin Qian, Yao Qian, Jian Wu, Michael Zeng, Xiangzhan Yu, and Furu Wei. 2022. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518.

Yushen Chen, Zhikang Niu, Ziyang Ma, Keqi Deng, Chunhui Wang, JianZhao JianZhao, Kai Yu, and Xie Chen. 2025. 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)*, pages 6255–6271.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. 2022. High fidelity neural audio compression. *arXiv preprint arXiv:2210.13438*.

Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. 2020. ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification. In *Interspeech 2020*, pages 3830–3834.

Zhihao Du, Changfeng Gao, Yuxuan Wang, Fan Yu, Tianyu Zhao, Hao Wang, Xiang Lv, Hui Wang, Chongjia Ni, Xian Shi, and 1 others. 2025. Cosyvoice 3: Towards in-the-wild speech generation via scaling-up and post-training. *arXiv preprint arXiv:2505.17589*.

Zhihao Du, Yuxuan Wang, Qian Chen, Xian Shi, Xiang Lv, Tianyu Zhao, Zhifu Gao, Yexin Yang, Changfeng Gao, Hui Wang, and 1 others. 2024. Cosyvoice 2: Scalable streaming speech synthesis with large language models. *arXiv preprint arXiv:2412.10117*.

Sefik Emre Eskimez, Xiaofei Wang, Manthan Thakker, Canrun Li, Chung-Hsien Tsai, Zhen Xiao, Hemin Yang, Zirun Zhu, Min Tang, Xu Tan, and 1 others. 2024. E2 tts: Embarrassingly easy fully non-autoregressive zero-shot tts. In *2024 IEEE spoken language technology workshop (SLT)*, pages 682–689. IEEE.

Haorui He, Zengqiang Shang, Chaoren Wang, Xuyuan Li, Yicheng Gu, Hua Hua, Liwei Liu, Chen Yang, Jiaqi Li, Peiyang Shi, Yuancheng Wang, Kai Chen, Pengyuan Zhang, and Zhizheng Wu. 2024. Emilia: An extensive, multilingual, and diverse speech dataset for large-scale speech generation. *Preprint*, arXiv:2407.05361.

Jonathan Ho and Tim Salimans. 2022. Classifier-free diffusion guidance. *Preprint*, arXiv:2207.12598.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *Preprint*, arXiv:2106.07447.

Hangrui Hu, Xinfa Zhu, Ting He, Dake Guo, Bin Zhang, Xiong Wang, Zhifang Guo, Ziyue Jiang, Hongkun Hao, Zishan Guo, and 1 others. 2026. Qwen3-tts technical report. *arXiv preprint arXiv:2601.15621*.

Yuma Koizumi, Heiga Zen, Shigeki Karita, Yifan Ding, Kohei Yatabe, Nobuyuki Morioka, Michiel Bacchiani, Yu Zhang, Wei Han, and Ankur Bapna. 2023. Libritts-r: A restored multi-speaker text-to-speech corpus. *Preprint*, arXiv:2305.18802.

Pin-Jui Ku, He Huang, Jean-Marie Lemercier, Subham Sekhar Sahoo, Zhehuai Chen, and Ante Jukić. 2026. Discrete diffusion for generative modeling of text-aligned speech tokens. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 17022–17026. IEEE.

Matthew Le, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, Rashel Moritz, Mary Williamson, Vimal Manohar, Yossi Adi, Jay Mahadeokar, and 1 others. 2023. Voicebox: Text-guided multilingual universal speech generation at scale. *Advances in neural information processing systems*, 36:14005–14034.

Shufan Li, Konstantinos Kallidromitis, Hritik Bansal, Akash Gokul, Yusuke Kato, Kazuki Kozuka, Jason Kuen, Zhe Lin, Kai-Wei Chang, and Aditya Grover. 2026. Lavida: A large diffusion language model for multimodal understanding. *Advances in Neural Information Processing Systems*, 38:105101–105134.

Wenrui Liu, Qian Chen, Wen Wang, Yafeng Chen, Jin Xu, Zhifang Guo, Guanrou Yang, Weiqin Li, Xiaoda Yang, Tao Jin, Minghui Fang, Jialong Zuo, Jionghao Bai, and Zemin Liu. 2025. Speech token prediction via compressed-to-fine language modeling for speech generation. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM '25)*. ArXiv:2505.24496.

Aaron Lou, Chenlin Meng, and Stefano Ermon. 2024. Discrete diffusion modeling by estimating the ratios of the data distribution. In *ICML*.

Aleksandr Meister, Matvei Novikov, Nikolay Karpov, Evelina Bakhturina, Vitaly Lavrukhin, and Boris Ginsburg. 2023. Librispeech-pc: Benchmark for evaluation of punctuation and capitalization capabilities of end-to-end asr models. *Preprint*, arXiv:2310.02943.

Tu Anh Nguyen, Wei-Ning Hsu, Antony D'Avirro, Bowen Shi, Itai Gat, Maryam Fazel-Zarani, Tal Remez, Jade Copet, Gabriel Synnaeve, Michael Hassid, Felix Kreuk, Yossi Adi, and Emmanuel Dupoux. 2023. Expresso: A benchmark and analysis of discrete expressive speech resynthesis. *Preprint*, arXiv:2308.05725.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. 2026. Large language diffusion models. *Advances in Neural Information Processing Systems*, 38:50608–50646.

Titouan Parcollet, Yuan Tseng, Shucong Zhang, and Rogier van Dalen. 2025. Loquacious set: 25,000 hours of transcribed and diverse english speech recognition data for research and commercial use. *Preprint*, arXiv:2505.21578.

Vineel Pratap, Qiantong Xu, Anuroop Sriram, Gabriel Synnaeve, and Ronan Collobert. 2020. Mls: A large-scale multilingual dataset for speech research. In *Proc. Interspeech 2020*, pages 2757–2761.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision. *Preprint*, arXiv:2212.04356.

Resemble AI. 2025. Chatterbox-TTS. `https://github.com/resemble-ai/chatterbox`. GitHub repository.

Julius Richter, Yi-Chiao Wu, Steven Krenn, Simon Welker, Bunlong Lay, Shinji Watanabe, Alexander Richard, and Timo Gerkmann. 2024. Ears: An anechoic fullband speech dataset benchmarked for speech enhancement and dereverberation. *Preprint*, arXiv:2406.06185.

Junhao Ruan, Bei Li, Yongjing Yin, Pengcheng Huang, Xin Chen, Jingang Wang, Xunliang Cai, Tong Xiao, and JingBo Zhu. 2026. Causal autoregressive diffusion language model. *Preprint*, arXiv:2601.22031.

Takaaki Saeki, Detai Xin, Wataru Nakata, Tomoki Koriyama, Shinnosuke Takamichi, and Hiroshi Saruwatari. 2022. Utmos: Utokyo-sarulab system for voicemos challenge 2022. *Preprint*, arXiv:2204.02152.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T. Chiu, Alexander M. Rush, and Volodymyr Kuleshov. 2024. Simple and effective masked diffusion language models. In *NeurIPS*.

Kai Shen, Zeqian Ju, Xu Tan, Eric Liu, Yichong Leng, Lei He, Tao Qin, Jiang Bian, and 1 others. 2024. Naturalspeech 2: Latent diffusion models are natural and zero-shot speech and singing synthesizers. In *International conference on learning representations*, volume 2024, pages 698–722.

Yingte Shu, Yuchuan Tian, Chao Xu, Yunhe Wang, and Hanting Chen. 2026. Deferred commitment decoding for diffusion language models with confidence-aware sliding windows. *arXiv preprint arXiv:2601.02076*.

Amitay Sicherman and Yossi Adi. 2023. Analysing discrete self supervised speech representation for spoken language modeling. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing*.

Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno. 2020. Generalized end-to-end loss for speaker verification. *Preprint*, arXiv:1710.10467.

Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, Lei He, Sheng Zhao, and Furu Wei. 2023. Neural codec language models are zero-shot text to speech synthesizers. *Preprint*, arXiv:2301.02111.

Wenbin Wang, Yang Song, and Sanjay Jha. 2024. Globe: A high-quality english corpus with global accents for zero-shot speaker adaptive text-to-speech. *Preprint*, arXiv:2406.14875.

Yuancheng Wang, Haoyue Zhan, Liwei Liu, Ruihong Zeng, Haotian Guo, Jiachen Zheng, Qiang Zhang, Xueyao Zhang, Shunsi Zhang, and Zhizheng Wu. 2025. Maskgct: Zero-shot text-to-speech with masked generative codec transformer. In *International Conference on Learning Representations*, volume 2025, pages 47127–47150.

Chengyue Wu, Hao Zhang, Shuchen Xue, Shizhe Diao, Yonggan Fu, Zhijian Liu, Pavlo Molchanov, Ping Luo, Song Han, and Enze Xie. 2025a. Fast-dllm v2: Efficient block-diffusion llm. *Preprint*, arXiv:2509.26328.

Chengyue Wu and 1 others. 2025b. Fast-dllm: Training-free acceleration of diffusion llm by enabling kv cache and parallel decoding. *arXiv preprint arXiv:2505.22618*.

Yifan Yang, Bing Han, Hui Wang, Long Zhou, Wei Wang, Mingyu Cui, Xu Tan, and Xie Chen. 2026. Measuring prosody diversity in zero-shot tts: A new metric, benchmark, and exploration. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 17252–17256. IEEE.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. 2025a. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*.

Zihao Ye, Lequn Chen, Ruihang Lai, Wuwei Lin, Yineng Zhang, Stephanie Wang, Tianqi Chen, Baris Kasikci, Vinod Grover, Arvind Krishnamurthy, and Luis Ceze. 2025b. Flashinfer: Efficient and customizable attention engine for llm inference serving. *Preprint*, arXiv:2501.01005.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. 2021. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:495–507.

Tao Zewei and Huang Yunpeng. 2025. Magiattention: A distributed attention towards linear scalability for ultra-long context, heterogeneous mask training. https://github.com/SandAI-org/MagiAttention/.

Siyi Zhou, Yiquan Zhou, Yi He, Xun Zhou, Jinchao Wang, Wei Deng, and Jingchen Shu. 2026. Indextts2: A breakthrough in emotionally expressive and duration-controlled auto-regressive zero-shot text-to-speech. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 35139–35148.

Yixuan Zhou, Guoyang Zeng, Xin Liu, Xiang Li, Renjie Yu, Ziyang Wang, Runchuan Ye, Weiyue Sun, Jiancheng Gui, Kehan Li, and 1 others. 2025. Voxcpm: Tokenizer-free tts for context-aware speech generation and true-to-life voice cloning. *arXiv preprint arXiv:2509.24650*.

Han Zhu, Wei Kang, Zengwei Yao, Liyong Guo, Fangjun Kuang, Zhaoqing Li, Weiji Zhuang, Long Lin, and Daniel Povey. 2025. Zipvoice: Fast and high-quality zero-shot text-to-speech with flow matching. *arXiv preprint arXiv:2506.13053*.

Han Zhu, Lingxuan Ye, Wei Kang, Zengwei Yao, Liyong Guo, Fangjun Kuang, Zhifeng Han, Weiji Zhuang, Long Lin, and Daniel Povey. 2026. Omnivoice: Towards omnilingual zero-shot text-to-speech with diffusion language models. *arXiv preprint arXiv:2604.00688*.

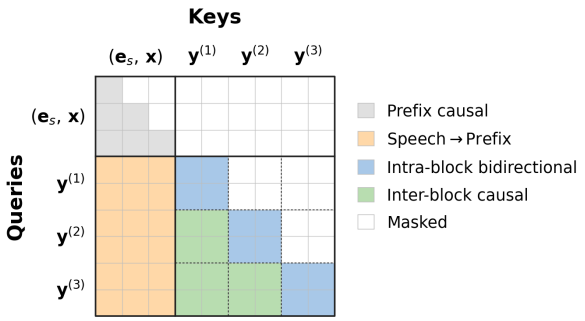

Figure 3: Hybrid block attention mask at $D = 2$. A colored cell at row i, column j indicates that query position i attends to key position j. The conditioning prefix $(\mathbf{e}_s, \mathbf{x})$ uses causal attention (gray); speech tokens attend to the entire prefix (orange); within each speech block $\mathbf{y}^{(b)}$, attention is bidirectional (blue); across blocks, attention is causal (green). Future blocks are never visible to past blocks, enabling block-by-block streaming.

A Attention Kernel Implementation

The hybrid attention mask described in Section 2.2 is implemented using either (i) PyTorch's native `flex_attention` with a custom mask function, or (ii) MagiAttention's Flex-Flash-Attention (FFA) (ai et al., 2025; Zewei and Yunpeng, 2025)[2] kernel for higher throughput on long sequences.

Range Encoding for FFA FFA expresses sparse attention as a list of rectangular query/key index ranges with per-range attention types (`FULL` or `CAUSAL`). Letting $L_{\text{pre}} = |\mathbf{e}_s| + |\mathbf{x}|$ denote the prefix length, N the speech-stream length, and D the block size, our hybrid mask decomposes into four rectangle groups:

1. **Prefix Self-Attention** (`CAUSAL`): $q, k \in [0, L_{\text{pre}})$.

[2] https://github.com/SandAI-org/MagiAttention

T	LibriSpeech-PC test-clean			Seed-TTS test-en		
	SIM-o↑	WER↓	UTMOS↑	SIM-o↑	WER↓	UTMOS↑
0.2 (default)	0.715	**1.61**	4.29	0.703	**1.75**	4.09
0.4	0.715	1.64	4.29	0.704	1.82	4.08
0.6	0.714	1.75	4.28	0.702	1.93	4.07
0.8	0.713	1.75	4.26	0.702	2.05	4.05
1.0	0.714	2.01	4.24	0.702	2.15	4.02

Table 5: Sampling temperature T sweep at $\beta = 5$, others at canonical default.

2. **Speech-to-Prefix** (FULL): $q \in [L_{\text{pre}},\ L_{\text{pre}}{+}N)$, $k \in [0,\ L_{\text{pre}})$.

3. **Intra-Block Bidirectional** (FULL): for each block b, $q, k \in [L_{\text{pre}}{+}bD,\ L_{\text{pre}}{+}(b{+}1)D)$.

4. **Inter-Block Left-Context** (FULL): for $b \geq 1$, q in block b, k in blocks $0, \ldots, b{-}1$.

Equivalence The boolean mask induced by the FFA range union is identical to the reference mask used by `flex_attention`; both backends produce numerically equivalent outputs up to bf16 noise.

Dataset	# Samples
Public	
MLS-English (Pratap et al., 2020)	10.8M
Emilia (en, part 1) (He et al., 2024)	9.1M
Loquacious (Parcollet et al., 2025)	3.9M
GLOBE (Wang et al., 2024)	582K
LibriTTS-R (Koizumi et al., 2023)	375K
HiFi-TTS (Bakhturina et al., 2021)	324K
EARS (Richter et al., 2024)	12K
Expresso (Nguyen et al., 2023)	12K
Privately collected	
Audiobook	17.7M
Podcasts	726K
IVR (VC-augmented)	445K
Short-form utterances†	292K
Conversational	50K
Stylized speech	62K
Total	**43.8M (∼70k hours)**

Table 6: Training dataset composition. Sample counts are reported after preprocessing and integrity filtering. †Short-form includes phone numbers, names, time expressions, and expressive single-word utterances.

B Training Dataset Details

Table 6 lists the datasets used for training. The total training set contains 43.8M utterances (∼70k hours) from 528k speakers, distributed across public and privately collected sources. The privately collected portion broadens coverage of conversational, expressive, and short-form utterances beyond standard read-speech corpora.

C Additional Ablations

This appendix reports additional ablations supplementing Section 3.5: sampling and position temperatures (Appendix C.1) and the CFG scale w (Appendix C.2). All sweeps use the canonical configuration of Section 3.2 except for the swept parameter.

C.1 Sampling and Position Temperatures

Sampling Temperature Table 5 sweeps the speech-token sampling temperature T at the canonical configuration ($D = 16$, $K = 10$, $\alpha = 0$, $w = 1.0$, $\beta = 5$). $T = 0.2$ minimizes mean WER on both benchmarks (1.61 on LibriSpeech-PC, 1.75 on Seed-TTS) and is our default; increasing T progressively degrades WER (+0.4 at $T = 1.0$) while leaving SIM-o and UTMOS within noise.

Position Temperature and Decoding Method For the position temperature β (Gumbel perturbation on the prior-calibrated scores; Section 2.3.3), we compared $\beta = 0$ (deterministic top-n_k) against $\beta = 5$ across both decoding methods (TS schedule and PMI) and step budgets $K \in \{2, 5, 8\}$ (Table 7). This sweep simultaneously serves as the head-to-head comparison between the TS schedule baseline and PMI referenced in Section 3.5. $\beta = 5$ consistently lowers WER over $\beta = 0$ on both benchmarks across all settings, with the effect most pronounced at low K (e.g., at $K = 2$: PMI WER drops from 11.82 to 5.17 on LibriSpeech-PC, 13.98 to 5.37 on Seed-TTS); the same trend holds for the TS schedule. Comparing the TS schedule baseline and PMI at $\beta = 5$, the two methods track each other closely across all step budgets—PMI is slightly ahead at $K = 5, 8$ on LibriSpeech-PC and at $K = 2$ on both benchmarks, while the TS schedule is slightly ahead at $K = 5$ on Seed-TTS—confirming that the choice of decoding method contributes only marginally to quality. PMI's empirical advantage manifests when early decoding is applied (e.g., $\alpha = 0.5$ in Table 1), where the calibrated confi-

K	Method	β	LibriSpeech-PC test-clean			Seed-TTS test-en		
			SIM-o↑	WER↓	UTMOS↑	SIM-o↑	WER↓	UTMOS↑
2	TS schedule	0	0.679	15.10	4.00	0.676	15.40	3.81
	TS schedule	5	0.697	8.74	4.13	0.690	9.13	3.93
	PMI	0	0.690	11.82	4.08	0.682	13.98	3.86
	PMI	5	0.707	**5.17**	**4.21**	0.697	**5.37**	**4.01**
5	TS schedule	0	0.714	2.52	4.26	0.702	2.50	4.08
	TS schedule	5	0.714	1.91	4.28	0.702	**2.13**	4.09
	PMI	0	0.715	2.36	4.27	0.703	2.46	4.07
	PMI	5	0.713	**1.96**	4.27	0.703	2.01	4.08
8	TS schedule	0	0.714	1.87	4.28	0.703	2.07	4.09
	TS schedule	5	0.714	1.69	4.29	0.703	1.97	4.09
	PMI	0	0.716	1.82	4.28	0.703	2.25	4.09
	PMI	5	**0.717**	**1.67**	**4.29**	**0.704**	**1.96**	4.09

Table 7: Position temperature β ablation at $D = 16$, $\tau = 0.5$, $w = 1.0$, $T = 0.2$. $\beta = 5$ (Gumbel-perturbed selection) consistently outperforms $\beta = 0$ (deterministic top-n_k) across both decoding methods (TS schedule, PMI) and all step budgets, with the effect most pronounced at low K. Best WER per K is in **bold**.

dence signal enables the per-step quantile threshold of Equation (9) to terminate blocks early without quality regression. SIM-o and UTMOS also improve modestly with $\beta = 5$. We adopt $\beta = 5$ throughout.

C.2 Classifier-Free Guidance Scale

Table 8 sweeps the CFG scale w at the canonical configuration ($D = 16$, $K = 10$, $\alpha = 0$, $T = 0.2$, $\beta = 5$). The two benchmarks favor slightly different operating points: LibriSpeech-PC attains its lowest WER at $w = 0.5$ (1.52), while Seed-TTS test-en attains its lowest at $w = 1.5$ (1.77). $w = 1.0$ provides the most balanced trade-off across both—1.63 / 1.79 WER with the highest Seed-TTS SIM-o (0.705)—and is our default. Larger w (≥ 1.5) marginally trades intelligibility against SIM-o and UTMOS on Seed-TTS.

D Streaming Server Quality

Table 9 reports objective quality metrics measured on the streaming inference server used for the latency and throughput numbers in Table 3. The streaming setup operates the model under chunk-wise vocoder decoding, with two production-oriented optimizations described below. Overall, the streaming setup exhibits a modest quality drop relative to the offline inference reported in our main results (Table 1): at the default configuration ($D = 16$, $\alpha = 0.5$), the streaming server attains SIM-o 0.688, WER 2.03, and UTMOS 4.07 on LibriSpeech-PC test-clean, drops of -0.025, $+0.36$, and -0.21 relative to offline. Across the six configurations evaluated, SIM-o and UTMOS remain stable while WER degrades at larger block sizes, consistent with the inference block-size ablation (Figure 2).

Vocoder Chunk Schedule The chunk-wise vocoder operates with a progressively widening chunk schedule. The first chunk has a duration of 0.46 s ($\sim$12 tokens at the 25 Hz codec rate), with each subsequent chunk grown by a factor of 5.0, capped at 6.0 s (150 tokens). A small initial chunk reduces TTFP, since the vocoder can emit the first audio packet as soon as a short window of speech tokens is available; subsequent larger chunks amortize vocoder overhead during sustained synthesis. The chunk duration can be tuned per deployment scenario, with shorter initial chunks reducing TTFP at the cost of slightly higher steady-state cost.

Early-Emit Block Serving We further reduce streaming latency through an *early-emit* optimization at the block boundary. During block decoding, masked positions within a block are unmasked progressively, and the early-decoding schedule (see Early Decoding paragraph) typically commits positions in a non-contiguous order. Whenever the committed positions form a contiguous left-aligned prefix of the block—that is, the remaining masked positions occupy only the right portion—we emit the prefix tokens to the vocoder immediately, without waiting for the entire block to finish decoding. This overlaps vocoder decoding of the early-committed tokens with the remaining denoising steps for the same block, yielding additional latency reduction while preserving the block-by-block streaming abstraction.

w	LibriSpeech-PC test-clean			Seed-TTS test-en		
	SIM-o↑	WER↓	UTMOS↑	SIM-o↑	WER↓	UTMOS↑
0.5	0.715	**1.52**	**4.30**	0.700	1.94	**4.12**
1.0 (default)	0.715	1.63	4.29	**0.705**	1.79	4.08
1.5	0.715	1.71	4.27	0.703	**1.77**	4.05
2.0	0.714	1.80	4.26	0.702	1.94	4.02

Table 8: CFG scale w sweep at the canonical configuration. $w = 1.0$ is the default used in our main results.

Config	SIM-o↑	WER↓	UTMOS↑
$D = 16, \alpha = 0.5$ (default)	0.688	2.03	4.07
$D = 16, \alpha = 0.75$	0.687	2.27	4.06
$D = 24, \alpha = 0.5$	0.687	2.63	4.07
$D = 24, \alpha = 0.75$	0.688	2.74	4.07
$D = 32, \alpha = 0.5$	0.686	3.86	4.06
$D = 32, \alpha = 0.75$	0.685	3.70	4.05
Offline (for reference, from Table 1)			
$D = 16, \alpha = 0.5$ (default)	0.713	1.67	4.28

Table 9: Streaming server quality on LibriSpeech-PC test-clean, measured under the same inference configurations as Table 3. The streaming setup exhibits a modest quality drop versus offline decoding (last row), most pronounced in WER at larger inference block sizes.

E Implementation Details

The inference engine is built on FlashInfer, with three customizations specific to block-diffusion TTS that differ from standard LLM serving.

Hybrid Causal/Non-Causal Attention Standard LLM serving uses causal attention throughout. Our inference loop instead interleaves two attention patterns within a single decoding step: the conditioning prefix is encoded with causal attention and cached once, while the current block is decoded with non-causal attention so that all masked positions in the block attend bidirectionally to each other (Section 2.3.1). The two patterns share the same paged key-value buffer and differ only in the causal flag passed to the attention kernel.

Frozen Prefix, Growing Block Cache The prefix forward is invoked once per generation and writes the prefix key-value entries into the paged buffer. Subsequent block forwards reuse this frozen prefix cache, append the current block's key-value entries, and never recompute prefix attention. This realizes the two-cache abstraction described in Section 2.3.1.

Cache Snapshot for CFG Classifier-free guidance (Section 2.3.4) requires both a conditional and an unconditional forward at each block step. We snapshot the paged cache before the unconditional pass and restore it afterwards, so the conditional state is preserved without recomputing the prefix.

CUDA Graph Replay Because the block size D is fixed at inference time, the per-step forward—attention, feed-forward, and the speech head—is captured as a single CUDA graph and replayed for every step of every block. This removes the per-launch overhead that otherwise dominates short-sequence, small-batch inference.

F Block-Size Scaling Explorations

This appendix summarizes alternative block-diffusion configurations we explored when attempting to scale beyond the block sizes used in our main results. None of these alternatives delivered both stable training at large block sizes and the prosodic quality required for production deployment, but they characterize where the difficulty lies and motivate future work.

Fully Causal Block Formulation (CARD-style) Following the causal autoregressive diffusion formulation of CARD (Ruan et al., 2026), we trained a variant in which all attention within and across blocks is strictly causal, removing the intra-block bidirectional context of our default formulation. At small block sizes ($D \leq 4$), this variant produces intelligible speech and offers the fastest inference among the configurations we tested, since a single forward pass per block is sufficient and no iterative refinement is required. Starting from $D = 5$, however, generated speech begins to exhibit prosodic collapse and over-smoothing, with reduced pitch and energy variation and noticeably flatter intonation. The fully causal training signal appears to be too weak to support meaningful parallelism inside the block, suggesting that the bidirectional intra-block context used by our main formulation is necessary for prosodic fidelity in TTS.

Block-Size Annealing with Self-Distillation Inspired by A2D-VL (Arriola et al., 2025a), which

gradually increases the diffusion prediction window during fine-tuning, we experimented with a data-free self-distillation recipe that anneals the block size starting from $D = 1$ (purely autoregressive) and progressively widening the block. This approach was able to maintain coherent speech up to roughly $D = 8$. Beyond this point, the model's per-position confidence collapses to a narrow range, and sampling becomes highly sensitive to temperature and CFG scale—small perturbations in the sampling configuration produce qualitatively different outputs. Extending the schedule to larger block sizes did not produce a stable regime within our compute budget.

Discussion Across these explorations, two distinct failure modes recur: (i) loss of prosodic variability when the training signal does not enforce bidirectional intra-block context, and (ii) collapsed confidence and high sampling sensitivity when block size grows faster than the model's denoising capability adapts. Our default block-diffusion formulation with bidirectional intra-block attention and a moderate block size offered the most stable trade-off among the recipes we tested, but a principled approach to scaling block size while preserving both prosody and confidence calibration remains an open question.

G Streaming Capability of Baselines

Our main results compare Chatterbox-Flash against NAR baselines that, to our knowledge, are not designed for streaming inference. In particular, OmniVoice (Zhu et al., 2026) is formulated as a full-sequence masked diffusion model whose architecture does not natively support streaming. The authors have indicated[3] that a form of *pseudo-streaming* can nonetheless be obtained by splitting the input text into smaller chunks and processing them sequentially.

We did not evaluate this configuration in the present work for two reasons. First, chunked sampling deviates from the model's training-time assumption of full-sequence bidirectional context, and the resulting quality–latency trade-off would likely depend on chunking strategy, overlap, and re-tuning of the denoising schedule. Second, any number reported under such an adaptation would reflect our specific reimplementation rather than the original system, making a fair comparison difficult.

We view a controlled study of pseudo-streaming OmniVoice—and, more broadly, of how full-sequence diffusion models can be retrofitted for streaming—as a meaningful direction for future work, as it would help disentangle whether the streaming advantage observed in our setting stems from the block-diffusion formulation itself or from training-time alignment with the streaming inference pattern.

[3] https://github.com/k2-fsa/OmniVoice/issues/6