# DELTA-TTS: Adapting Autoregressive Model into Diffusion Language Model for Text-to-Speech

**Junwon Moon[1], Yejin Lee[1], Seungbeom Kim[1], Hoseong Ahn[1], Sewoong Park[1], Heeseung Kim[2], Kyuhong Shim[1]**
[1]Sungkyunkwan University [2]University of Seoul
{mppn98, khshim}@skku.edu

## Abstract

Autoregressive (AR) text-to-speech (TTS) models generate discrete speech tokens sequentially, which makes inference slow and can degrade robustness, since local errors propagate to later positions and can escalate into hallucination. This limitation stems from their left-to-right AR commitment: each token must be determined before future speech-token context is available. However, such ordering is not an inherent requirement for TTS, since the model receives the full input text before synthesis. In this paper, we introduce DELTA-TTS, a lightweight LoRA-based adaptation framework that converts a pretrained AR TTS model into a discrete diffusion language model (dLLM) for confidence-ordered speech-token decoding. To better capture the local structure of speech, DELTA-TTS incorporates a convolution module that injects local acoustic context, together with a $1/t$-weighted training objective and a time-shifted inference schedule that together defer low-confidence positions to later steps. Trained on only 585 hours of LibriTTS, DELTA-TTS achieves a 1.75% WER on Seed-TTS test-en, outperforming its AR backbone while generating tokens 3.3 times faster. Further analysis shows that DELTA-TTS produces sharper text–speech alignment, increases overall decoding confidence, and mitigates the hallucinations observed in AR generation. Audio samples, inference code, and model weights are available at `https://slpr-lab.github.io/DELTA-TTS`.

## 1 Introduction

Recent autoregressive (AR) text-to-speech (TTS) models synthesize highly natural speech using large language model (LLM) backbones that predict discrete speech tokens [11, 12, 22, 43]. Despite their strong performance, AR TTS models have two fundamental limitations. First, token-by-token generation leads to slow inference. This cost is particularly pronounced for speech, which requires far more tokens than text to represent the same content. Second, unidirectional generation conditions each token only on past context. Because neighboring speech tokens strongly depend on one another, the AR model's inability to attend to future context becomes a structural weakness.

Formally, AR TTS models the conditional distribution $P(A \mid T) = P(a_1, a_2, \cdots \mid T)$, where $A$ is a speech-token sequence and $T$ is the input text. The AR formulation factorizes this distribution as $P(a_1 \mid T)\, P(a_2 \mid a_1, T) \cdots$, forcing a fixed left-to-right generation order. However, because the entire text $T$ is given in advance, this factorization is not the only option for producing $A$. Since the learned model can only approximate the target conditional distribution, the decoding order affects how errors accumulate. We hypothesize that decoding tokens in order of confidence, rather than in temporal (i.e., left-to-right) order, leads to more stable generation. Interestingly, this hypothesis aligns closely with the generation process of discrete diffusion language models (dLLMs), which have recently emerged as a promising alternative to AR generation in the text domain [31, 34, 35, 42].

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

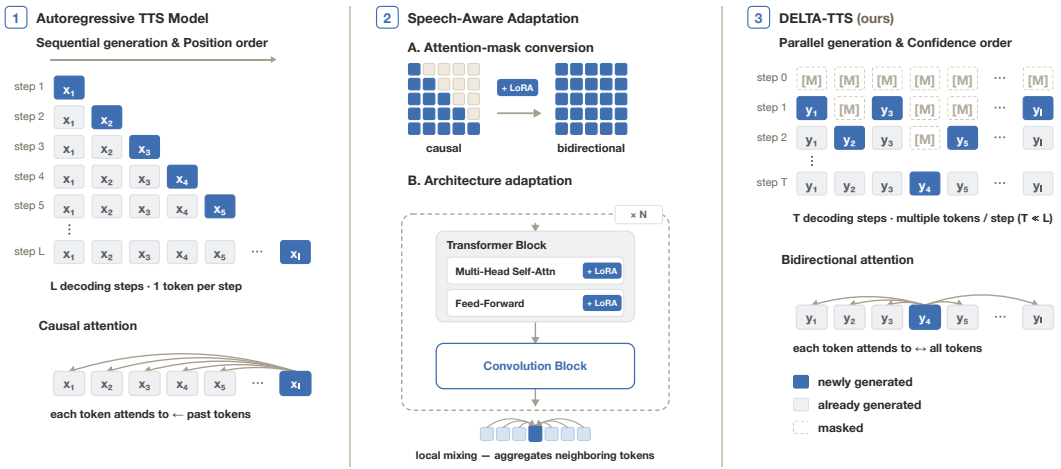



Figure 1: Overview of DELTA-TTS. A pretrained AR TTS backbone is converted into a discrete diffusion language model with two adaptations: bidirectional attention and block-wise LoRA with a Conformer-style convolution module.

In essence, dLLMs generate high-confidence tokens first and progressively complete the sequence through iterative refinement. In this paper, we propose **DELTA-TTS**, a framework that adapts a pretrained AR TTS model into a dLLM, generating speech non-autoregressively while preserving the synthesis quality of the backbone. The conversion process employs parameter-efficient LoRA-based modifications. To better reflect the local and bidirectional structure of speech, we introduce two additional refinements. First, we incorporate a Conformer-style convolution module to model the correlations among adjacent speech tokens. Second, we combine a $1/t$-weighted training objective with a time-shifted inference schedule so that tokens with insufficient context can be deferred to later decoding steps. With these refinements, the model learns a sharper text–speech alignment and reduces error accumulation caused by committing tokens too early.

We validate DELTA-TTS through both empirical evaluation and analysis. Trained on only 585 hours of LibriTTS, DELTA-TTS achieves 1.75% WER on Seed-TTS *test-en* while generating tokens 3.3× faster, and our analysis shows how each refinement contributes to these gains.

Our contributions are as follows:

- We introduce DELTA-TTS, the first method to extend AR-to-dLLM conversion to speech. DELTA-TTS converts a pretrained AR TTS backbone into a non-autoregressive model through a lightweight LoRA-based modification that requires only ∼15% additional parameters.
- We propose two speech-aware designs for generation in confidence order rather than temporal order: a convolution module for locality, and a $1/t$-weighted training objective with a time-shifted schedule for deferring uncertain tokens to later steps.
- We analyze the conversion from two perspectives. First, naive text-domain conversion substantially degrades performance, and the convolution module recovers it by sharpening the text–speech alignment. Second, confidence-ordered decoding maintains high confidence across the sequence, mitigating hallucination.

## 2 Related Work

### 2.1 Autoregressive and Non-Autoregressive TTS

Modern zero-shot AR TTS systems formulate speech synthesis as language modeling over discrete tokens and are trained on large speech corpora [1, 6, 11]. A growing line of work initializes their backbones from pretrained text language models [12, 37, 41]. On top of these backbones, they attach speech embedding and output projection layers to process discrete speech tokens. However, this

paradigm inherits two limitations from AR language modeling. First, generating an $N$-token speech sequence demands $N$ sequential forward passes, so inference latency grows linearly with the output length. Second, attending only to past context prevents the model from exploiting future tokens, despite the strong local dependencies among neighboring speech tokens.

NAR TTS models instead generate all tokens within a fixed number of refinement steps, independent of the output length. This parallel formulation is also naturally compatible with bidirectional attention, so the model can capture full-sequence context. These methods can be broadly categorized into continuous-space approaches [7, 13, 25–27] and discrete-token approaches [4, 24, 38, 39]. However, as with AR systems, NAR TTS models are typically trained from scratch on large-scale speech corpora, which incurs substantial computational cost.

To bridge the gap between these two paradigms, hybrid approaches combining AR and NAR generation have also been explored. For example, DiTAR [23] performs autoregressive generation at the chunk level while employing diffusion-based decoding within each chunk. While bidirectional attention operates within each chunk, generation across chunks remains sequential. As a result, the latency bottleneck is only partially resolved.

### 2.2 Discrete Diffusion Language Models

Diffusion-based language modeling has emerged as a competitive NAR paradigm. Early continuous-diffusion approaches [9, 16, 19, 28] suffer from scalability issues and information loss when mapping continuous representations back to discrete tokens. Subsequent work has therefore shifted toward discrete diffusion models that operate directly in the token space [3, 29, 34].

Specifically, these models learn to predict masked tokens in a partially masked sequence, reversing a forward masking process. At inference time, starting from a fully masked sequence, the model iteratively unmasks tokens in parallel and commits the most confident predictions at each step. Building on this paradigm, recent dLLMs [2, 31, 42, 47] have begun to narrow the quality gap with AR LLMs while retaining the efficiency advantages of parallel generation.

### 2.3 AR-to-dLLM Conversion

A recent line of research has shown that pretrained AR language models can be converted into dLLMs [15, 36, 40, 42]. These methods adapt pretrained AR weights through continual pretraining with a masked diffusion objective [34]. Many of these methods further incorporate a shift operation that preserves the next-token prediction behavior of the AR model, so the pretrained weights transfer directly to parallel generation. Because this conversion route avoids training from scratch, it substantially improves training efficiency. However, existing approaches have been developed primarily for text language models, and extending AR-to-dLLM conversion to TTS models remains unexplored.

## 3 Method

### 3.1 Overview

DELTA-TTS adapts a pretrained AR TTS model into a dLLM, while keeping the speech tokenizer and the flow-matching decoder frozen so that the existing zero-shot inference pipeline can be reused without modification. We use CosyVoice3 [11] as the AR backbone, in which a Qwen2.5-0.5B language model produces a 25 Hz semantic speech-token sequence and a flow-matching decoder converts these tokens into waveforms.

We adapt the backbone through two changes (Figure 1). First, we replace the causal self-attention mask with a bidirectional one. Second, we add two block-wise modules, a LoRA and a Conformer-style convolution module, on top of the frozen AR backbone. We detail these changes in Sections 3.2 and 3.3, followed by the training objective in Section 3.4 and the inference procedure in Section 3.5.

### 3.2 Bidirectional Adaptation

Following CosyVoice3's zero-shot input format, each sample is laid out as

$$[\mathrm{SOS}, \mathbf{t}_{\mathrm{inst}}, \mathbf{t}_{\mathrm{prompt}}, \mathbf{t}_{\mathrm{target}}, \mathrm{TASK}, \mathbf{s}_{\mathrm{prompt}}, \mathbf{s}_{\mathrm{target}}, \mathrm{EOS}], \tag{1}$$

where $\mathbf{t}_{\text{inst}}$, $\mathbf{t}_{\text{prompt}}$, and $\mathbf{t}_{\text{target}}$ denote the instruction, prompt, and target text tokens, and $\mathbf{s}_{\text{prompt}}$ and $\mathbf{s}_{\text{target}}$ denote the prompt and target speech tokens. Masking is applied only to $\mathbf{s}_{\text{target}}$.

During training, a random subset of its positions is replaced with the mask token $\mathrm{M}$, and the model learns to recover the original tokens. At inference time, $\mathbf{s}_{\text{target}}$ is initialized as a sequence of $\mathrm{M}$ tokens.

The AR backbone processes the resulting sequence under bidirectional self-attention. As a result, each masked target token has direct access to the surrounding speech context, the full transcript, and the prompt speech tokens that carry speaker identity.

### 3.3 Lightweight Adapter Architecture

We freeze the AR backbone and route all adaptation through two trainable modules attached to every transformer block. The first module is LoRA [21], which adds low-rank updates to the attention and feed-forward projections of each block while leaving the original weights untouched. Because the pretrained weights are not modified, the backbone retains its language modeling capabilities and speech-token representations while transitioning from causal to bidirectional attention. The low-rank updates alone are sufficient to support the newly introduced bidirectional context. In prior AR-to-dLLM adaptation [15], the bidirectional mask must be phased in gradually over training to prevent disruption of the fully trained backbone. Our approach removes the need for this mask-annealing schedule.

The second module is a Conformer-style convolution module [17], inserted as a residual branch after each transformer block and composed of a depth-wise convolution, GLU gating, and a Swish activation. Speech tokens at 25 Hz exhibit strong temporal locality. Neighboring positions encode acoustically similar frames, and most of the structure relevant to prosody and phoneme transitions lies within a short window. Bidirectional self-attention mixes information across the full sequence but has limited inductive bias toward local context. The convolution module compensates for this limitation with an explicit local-mixing operation, so the two mechanisms jointly cover both global structure and short-range continuity.

### 3.4 Training

We follow the masked discrete diffusion formulation [31, 34]. Let $\mathbf{s}_0$ denote the original target speech tokens. For each training example, we sample $t \in (0, 1)$. We then form a partially masked sequence $\mathbf{s}_t$ by replacing each token of $\mathbf{s}_0$ with the mask token $\mathrm{M}$ independently with probability $t$. All other tokens in the input (text, prompt speech, special tokens) remain visible and form the context $\mathbf{c}$. The model is trained to predict the original token at each masked position:

$$\mathcal{L}(\theta) \triangleq -\mathbb{E}_{t,\mathbf{s}_0,\mathbf{s}_t}\left[\frac{1}{t}\sum_{i=1}^{L}\mathbf{1}[\mathbf{s}_t^i = \mathrm{M}]\,\log p_\theta(\mathbf{s}_0^i \mid \mathbf{c}, \mathbf{s}_t)\right], \tag{2}$$

where $L$ is the target speech length and the indicator restricts the loss to masked positions. The $1/t$ weighting follows from the ELBO of the masked diffusion process [15, 31], amplifying the loss at low masking ratios where the model should predict with high confidence.

The AR backbone predicts the token at position $i{+}1$ from the hidden state at position $i$, whereas standard masked diffusion predicts the masked token at the same position $i$. We adopt the shift operation from prior AR-to-dLLM conversions [15, 42]: the hidden state at position $i$ predicts $\mathbf{s}_0^{i+1}$ rather than $\mathbf{s}_0^i$, preserving the AR backbone's next-token prediction behavior.

The mask token $\mathrm{M}$ is the only newly added embedding; we initialize it as the mean of all speech-token embeddings and let training refine it.

### 3.5 Inference

We decode $\mathbf{s}_{\text{target}}$ through $T$ iterations of parallel unmasking. At each iteration, the model predicts every masked position simultaneously and draws each candidate with top-$p$ nucleus sampling [20]. We then accept the candidates with the highest confidence, defined as the probability of the sampled token, and keep the rest masked for the next iteration.

Instead of unmasking a uniform number of tokens per step, we use a time-shifted schedule that adapts the continuous formulation of [14] to the discrete unmasking setting. Let $c_n$ be the cumulative

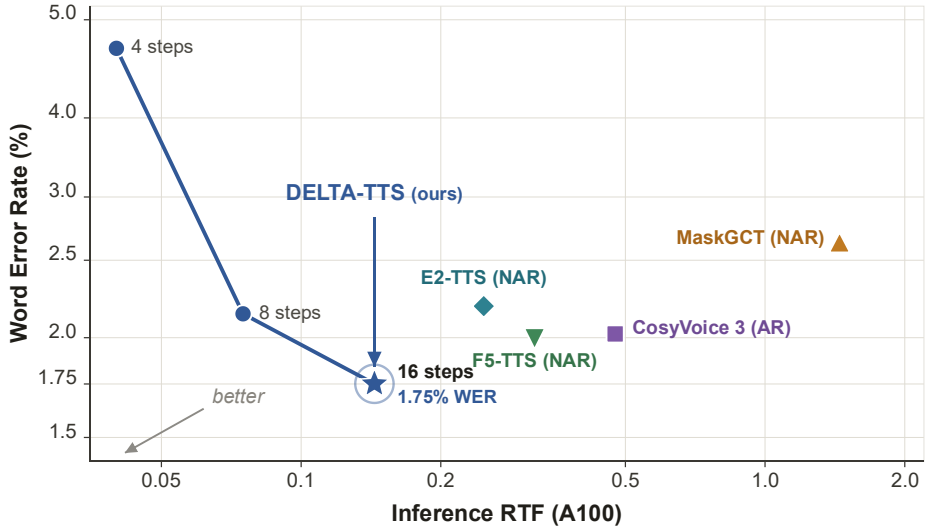



Figure 2: Speed–quality Pareto front on Seed-TTS *test-en* (both axes log-scale). DELTA-TTS at 16 steps achieves the lowest WER and RTF among all evaluated systems, offering the most favorable speed–quality trade-off.

fraction of tokens unmasked after step $n$ out of $T$:

$$c_n = \frac{\mu \cdot n/T}{1 + (\mu - 1) \cdot n/T}, \tag{3}$$

where $\mu > 0$ is a time-shifting parameter, and the number of decisions made at step $n$ is $k_n = \lfloor c_n L \rfloor - \lfloor c_{n-1} L \rfloor$. With $\mu < 1$, the schedule unmasks few tokens at early steps and concentrates most decisions toward the end.

This schedule creates a confidence-to-context decoding trajectory. When the sequence is still highly masked, the model commits only a few tokens, selecting the most confident positions first. Once committed, these tokens become reliable anchors that provide local and bidirectional speech context for the remaining masked positions. As decoding proceeds, the accumulated high-confidence tokens help resolve later, less confident positions under richer context. The time-shifted schedule in Eq. (3) avoids unreliable early commitments, and the few confident decisions made early support later, more difficult ones.

# 4 Experiments

## 4.1 Experimental Setup

### 4.1.1 Training and Evaluation Datasets

We train our AR-to-dLLM conversion on LibriTTS [44] (585 h of English speech). We evaluate on two zero-shot TTS benchmarks: Seed-TTS *test-en* [1] (1088 samples) from the Seed-TTS evaluation suite, and the LibriSpeech-PC *test-clean* Subset B [30] (1127 samples) released by F5-TTS [7].

### 4.1.2 Evaluation Metrics

We report four objective metrics. For Word Error Rate (WER), which assesses generation intelligibility, we follow prior conventions and use a different ASR system per benchmark: Whisper-Large-V3 [32] for Seed-TTS *test-en* and Faster-Whisper-Large-V3 for LibriSpeech-PC Subset B. For speaker similarity (SIM), we compute the cosine similarity between the generated and prompt speech using an ECAPA-TDNN model [8] based on WavLM-large [5]. We adopt UTMOS [33], a predicted mean opinion score, to assess speech naturalness objectively. We measure inference speed as the Real-Time Factor (RTF) on a single NVIDIA A100.

Table 1: Objective evaluation on Seed-TTS *test-en*. Baseline results are from official checkpoints or papers. Best results are in **bold** and second-best are underlined. ↑: higher is better; ↓: lower is better.

| Type | Model | Params. | Training Data (hours) | Seed-TTS *test-en* | |
|---|---|---|---|---|---|
| | | | | WER ↓ | SIM ↑ |
| – | **Ground-truth** | – | – | 2.14 | 0.734 |
| AR | Seed-TTS | N/A | N/A | 2.25 | **0.762** |
| | CosyVoice | 0.3B | 170K Multi. | 4.29 | 0.609 |
| | CosyVoice2 | 0.5B | 170K Multi. | 2.57 | 0.659 |
| | Spark TTS | 0.5B | 100K Multi. | 1.98 | 0.573 |
| | FireRedTTS2 | 1.5B | 1400K Multi. | 1.95 | 0.665 |
| | IndexTTS2 | 1.5B | 55K Emilia | 2.23 | 0.706 |
| | VoxCPM | 0.5B | 1800K Multi. | <u>1.85</u> | <u>0.729</u> |
| | Llasa | 1B | 250K Multi. | 3.22 | 0.572 |
| | CosyVoice3 | 0.5B | 1000K Multi. | 2.02 | 0.692 |
| NAR | MaskGCT (50 NFE) | 1.1B | 100K Emilia | 2.62 | 0.714 |
| | E2 TTS (32 NFE) | 0.3B | 100K Emilia | 2.19 | 0.710 |
| | F5-TTS (32 NFE) | 0.3B | 100K Emilia | 2.00 | 0.647 |
| Ours | DELTA-TTS | 0.5B+94M | 0.585K LibriTTS | **1.75** | 0.688 |

Table 2: Subjective evaluation on Seed-TTS *test-en*. Means with 95% confidence intervals.

| Model | CMOS ↑ | SMOS ↑ |
|---|---|---|
| Ground-truth | 0.00 | $3.80 \pm 0.09$ |
| CosyVoice3 | $-0.25 \pm 0.12$ | $4.00 \pm 0.08$ |
| DELTA-TTS | $+\mathbf{0.16} \pm 0.11$ | $\mathbf{4.14} \pm 0.07$ |

Table 3: Length-bucketed WER and RTF on Seed-TTS *test-en*. RTF is measured on the token-generation stage.

| Duration | WER (%) ↓ | | RTF ↓ | | Speedup ↑ |
|---|---|---|---|---|---|
| | CosyVoice3 | DELTA | CosyVoice3 | DELTA | |
| 0–3 s | 1.54 | **1.24** | 0.479 | **0.230** | 2.08× |
| 3–5 s | 1.74 | **1.45** | 0.475 | **0.144** | 3.30× |
| 5–10 s | 2.80 | **2.60** | 0.473 | **0.106** | 4.46× |
| All | 2.02 | **1.75** | 0.475 | **0.144** | 3.30× |

For subjective evaluation, we collect the comparative mean opinion score (CMOS, $[-3, 3]$) for side-by-side naturalness comparison against human audio, and the similarity mean opinion score (SMOS, $[1, 5]$) for speaker similarity to the prompt audio.

#### 4.1.3 Baselines

We compare **DELTA-TTS** against representative AR and NAR zero-shot TTS systems. AR baselines include Seed-TTS [1], CosyVoice [10], CosyVoice2 [12], Spark TTS [37], FireRedTTS [18], FireRedTTS2 [41], IndexTTS2 [45], Llasa [43], VoxCPM [46], DiTAR [23], and our base model CosyVoice3 [11]; NAR baselines include MaskGCT [38], E2 TTS [13], and F5-TTS [7]. For Seed-TTS *test-en*, baseline numbers are taken from the CosyVoice HuggingFace evaluation page[1] where available, and from the Llasa paper [43] for E2 TTS and Llasa.

For LibriSpeech-PC Subset B, baseline numbers are taken from F5-TTS [7] and DiTAR [23] where available, or measured by us using the released official checkpoints. For CosyVoice3 in particular, we measure WER, SIM, and UTMOS ourselves on both benchmarks using the released checkpoint, to ensure consistency with the rest of the evaluation pipeline.

[1] `https://huggingface.co/FunAudioLLM/CosyVoice-300M#evaluation`

Table 4: Objective evaluation on LibriSpeech-PC *test-clean* (Subset B, 1127 samples). Baseline results are from official checkpoints or papers. Best results are in **bold** and second-best are underlined. ↑: higher is better; ↓: lower is better.

| Type | Model | Params. | Training Data (hours) | LibriSpeech-PC *test-clean* | | |
|---|---|---|---|---|---|---|
| | | | | WER ↓ | SIM ↑ | UTMOS ↑ |
| – | **Ground-truth** | – | – | 2.23 | 0.69 | 4.10 |
| AR | CosyVoice | 0.3B | 170K Multi. | 3.59 | 0.66 | 4.14 |
| | FireRedTTS | 0.6B | 248K Multi. | 2.69 | 0.47 | – |
| | DiTAR | 0.6B | 100K Emilia | 2.39 | <u>0.67</u> | 4.22 |
| | CosyVoice3 | 0.5B | 1000K Multi. | **1.95** | **0.69** | **4.28** |
| NAR | MaskGCT (50 NFE) | 1.1B | 100K Emilia | 2.72 | **0.69** | 3.90 |
| | E2 TTS (32 NFE) | 0.3B | 100K Emilia | 2.95 | **0.69** | 3.56 |
| | F5-TTS (32 NFE) | 0.3B | 100K Emilia | 2.42 | 0.66 | 3.88 |
| Ours | DELTA-TTS | 0.5B+94M | 0.585K LibriTTS | <u>2.17</u> | **0.69** | <u>4.27</u> |

## 4.2 Main Results

### 4.2.1 Intelligibility

On Seed-TTS *test-en* (Table 1), DELTA-TTS achieves the lowest WER among all compared systems despite training on only 585 h of English speech. On LibriSpeech-PC Subset B (Table 4), DELTA-TTS again achieves a lower WER than every NAR baseline and is second only to the strongest AR system, showing that the conversion preserves the synthesis capability of the pretrained backbone. These results are obtained by training only $94$M parameters on top of the frozen backbone, without retraining the AR weights.

### 4.2.2 Speaker Similarity and Naturalness

DELTA-TTS achieves SIM competitive with leading AR and NAR baselines on both benchmarks, and UTMOS on Subset B is on par with the strongest AR systems, indicating that the parallel paradigm does not degrade perceptual audio quality. Subjective evaluation in Table 2 reinforces this finding. DELTA-TTS is judged more natural than ground truth in CMOS ($+0.16$, against $-0.25$ for CosyVoice3) and obtains the highest SMOS ($4.14$ vs. $4.00$ for CosyVoice3 and $3.80$ for ground truth), with both gaps over CosyVoice3 statistically significant (Appendix A.7).

## 4.3 Speed–Quality Trade-off

Figure 2 reports the speed–quality Pareto front on Seed-TTS *test-en*, obtained by sweeping the number of unmasking steps over $\{4, 8, 16\}$. With too few steps the masked positions are under-resolved and WER rises sharply, while quality improves steadily as the step budget grows. At $16$ steps, DELTA-TTS reaches the bottom-left corner of the front, and we adopt this setting as our default. DELTA-TTS then achieves a WER of $1.75\%$ at an RTF of $0.144$, a $3.3\times$ speedup over the AR baseline, and dominates the NAR baselines on both axes. RTF here is measured on the token-generation stage alone; end-to-end latency including the flow-matching vocoder is reported in Appendix A.2, where the speedup is $2.71\times$ at batch $1$.

Table 3 further shows that DELTA-TTS's fixed unmasking budget produces a larger speedup on longer utterances ($2.08\times \rightarrow 4.46\times$ from $< 3$ s to $\geq 5$ s) while retaining the WER advantage at every length.

## 4.4 Ablation Study

Table 5 breaks down each component of our adaptation. Adding the time-shifted schedule reduces WER over the uniform-schedule baseline, confirming the benefit of deferring uncertain commitments to later steps. The block-wise convolution module further reduces WER by introducing short-range mixing that complements bidirectional attention. Raising the LoRA rank to a comparable trainable

Table 5: Ablation of DELTA-TTS on Seed-TTS *test-en*. Rows marked "+" are cumulative on top of the preceding row. The two controls isolate trainable capacity and English-only adaptation, respectively, from the conversion itself.

| **Configuration** | **WER** (%) ↓ | **SIM** ↑ | **UTMOS** ↑ | **Trainable** |
|---|---|---|---|---|
| *Cumulative adaptation* | | | | |
| Naive conversion (LoRA, no Conv, uniform sched.) | 3.01 | 0.574 | 4.04 | 35M |
| + Time-shifted schedule | 2.59 | 0.586 | 4.05 | 35M |
| + Convolution module ($k$=31) | **1.61** | 0.584 | **4.09** | 94M |
| + Prompt conditioning (GT length) | 1.63 | 0.686 | 4.01 | 94M |
| + Prompt conditioning (rule-based length) | 1.75 | **0.688** | 3.97 | 94M |
| *Capacity-matched control* | | | | |
| LoRA only ($r$=160, no Conv) | 2.13 | 0.576 | 4.02 | 88M |
| *Control: adaptation without conversion* | | | | |
| LoRA AR fine-tuning (causal) | 1.87 | 0.687 | 3.98 | 35M |
| *Alternative training strategies* | | | | |
| Full fine-tuning | 1.97 | 0.680 | 4.01 | 494M |
| KD from CosyVoice3 | 2.37 | 0.591 | 4.05 | 94M |

Table 6: Backbone transfer on Seed-TTS *test-en*. The same adapter recipe is applied to two additional AR backbones without further tuning.

| **Model** | **WER** (%) ↓ | **SIM** ↑ |
|---|---|---|
| Llasa (AR backbone) | 3.72 | **0.519** |
| + DELTA-TTS | **3.13** | 0.454 |
| CosyVoice2 (AR backbone) | 2.50 | 0.602 |
| + DELTA-TTS | **2.31** | **0.614** |

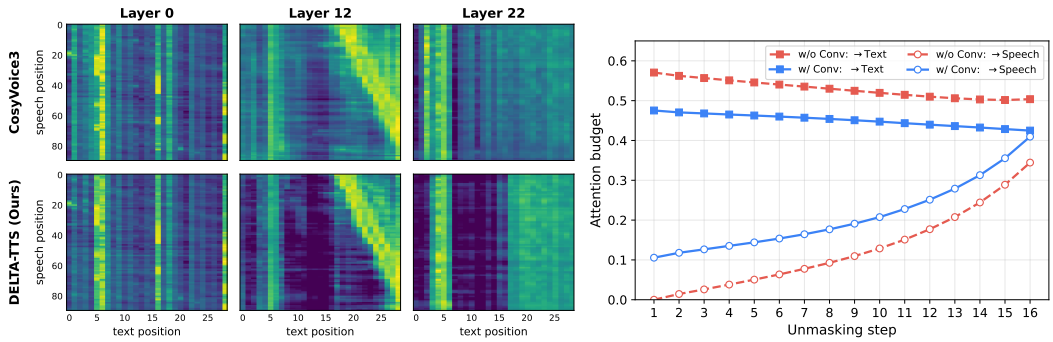
budget without the convolution reduces WER only to $2.13\%$, so the gain comes from the convolutional inductive bias rather than added capacity. Prompt-conditioned training trades a small amount of WER and naturalness for a substantial gain in speaker similarity.

Fully unfreezing the backbone increases WER despite more parameters, suggesting that the limited 585-hour adaptation data is insufficient to fine-tune all 494M parameters without overfitting. Parameter-efficient adapters are therefore preferable in this low-resource setting. Fine-tuning the AR backbone on the same corpus with LoRA while keeping causal decoding reaches $1.87\%$, so part of the gain comes from English-only adaptation; the conversion accounts for the remainder as well as the entire speedup. A knowledge-distillation baseline that trains the same dLLM to match CosyVoice3's outputs lags our direct conversion in both WER and SIM, showing that distilling from a teacher caps the student at the teacher's quality, whereas direct conversion can surpass the AR backbone.

For the rule-based length variant, we use the length estimation rule of OmniVoice [48], which sets the target speech-token count from the audio-to-character rate of the prompt and the weighted character count of the target text.

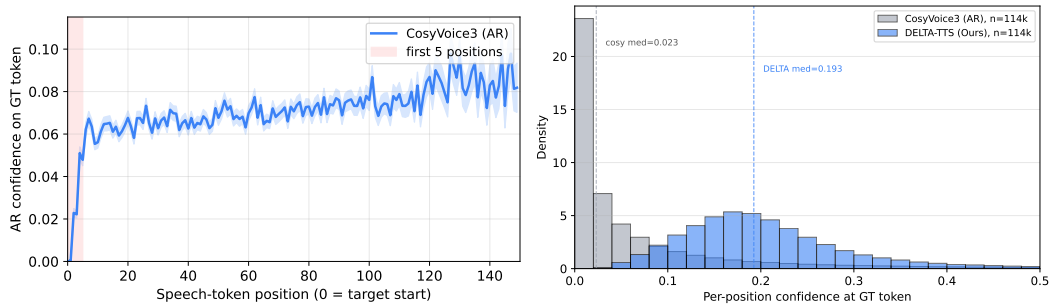
### 4.5 Backbone Transfer

The ablations above are all measured on a single backbone, so we test whether the conversion recipe itself is backbone-agnostic by applying it to two additional AR TTS systems, Llasa (1B) and CosyVoice2, keeping the same LoRA rank, the same per-block convolution, and the same LibriTTS adaptation data. For Llasa, whose codec produces $50$ Hz tokens and therefore roughly twice the sequence length of CosyVoice3, we adjust only the decoding schedule to $32$ unmasking steps; CosyVoice2 uses our CosyVoice3 configuration unchanged.




(a) Text–speech alignment (b) Attention budget of masked tokens

Figure 3: (a) Text–speech alignment across depth: DELTA-TTS sharpens the middle-layer diagonal and, in the late layer, attends to upcoming text, a pattern unavailable to the causal AR baseline. (b) Per-step attention budget from masked tokens; the convolution module shifts more attention onto neighboring unmasked speech while keeping text attention stable.




(a) AR confidence by token position (b) Marginal confidence distribution

Figure 4: (a) Mean teacher-forced AR confidence at the GT speech token across 1088 Seed-TTS *test-en* utterances; the first 5 positions sit close to zero. (b) Marginal distribution of per-position confidence at the GT speech token; the AR model's teacher-forced confidence concentrates near zero, while DELTA-TTS commits at far higher confidence.

The conversion improves WER on both backbones (Table 6), and on CosyVoice2 it improves speaker similarity as well, indicating that the recipe does not depend on the particular tokenizer or backbone it was developed on.

### 4.6 Analysis

We analyze DELTA-TTS from two angles. We first examine how the conversion reshapes attention over text and speech, and then how confidence-ordered decoding mitigates the hallucinations that afflict AR generation.

#### 4.6.1 Text–Speech Alignment and Speech-Token Locality

Figure 3(a) shows that DELTA-TTS produces a substantially cleaner text–speech alignment than CosyVoice3, with the middle-layer diagonal sharpening from a noisy blur into a clearly defined line. This sharpening does not come from the adaptation data but from the convolution module: a LoRA fine-tune on the same corpus with causal attention does not sharpen at all (Appendix A.3).

Figure 3(b) shows that masked positions concentrate more attention on neighboring unmasked speech tokens while attention to text remains stable, and this strengthened local focus tightens the alignment. Beyond this, the late layer attends to upcoming text, exploiting future context that is structurally inaccessible to the causal AR decoder.

Table 7: Tail of the per-utterance WER distribution on Seed-TTS *test-en* over $5440$ generations per system ($5$ seeds $\times$ $1088$ utterances). The AR backbone abandons the target text entirely on 5 generations, whereas DELTA-TTS never does.

| | CosyVoice3 (AR) | DELTA-TTS |
|---|---|---|
| Complete collapse (WER $\geq 80\%$) | $5$ | $\mathbf{0}$ |
| WER $\geq 50\%$ | $7$ | $\mathbf{3}$ |
| Worst-case per-utterance WER | $100\%$ | $\mathbf{64}\%$ |
| Perfect transcription (WER $= 0$) | $83.9 \pm 0.3\%$ | $\mathbf{86.2 \pm 0.8}\%$ |

#### 4.6.2 Confidence Advantage and Robustness to AR Failure Modes

Autoregressive decoders must commit each token before observing the tokens that follow, so early positions are decided under insufficient context and the resulting errors propagate. DELTA-TTS removes this constraint. Bidirectional attention grants every position access to future context, and confidence-ordered decoding defers the positions that require more context until enough has accumulated.

Figure 4(a) plots the AR model's confidence at the ground-truth (GT) speech token at each position. The first 5 positions sit close to zero: the AR model places almost no probability mass on the GT token at the utterance start, yet must commit these positions from the prompt context alone. DELTA-TTS instead defers them through confidence-ordered decoding (Appendix A.10). Figure 4(b) aggregates these per-position confidences into a marginal distribution. The AR model's teacher-forced confidence concentrates near zero (median $0.023$), whereas DELTA-TTS commits at far higher confidence (median $0.193$, $8\times$ higher), with negligible mass in that region.

This confidence advantage has a measurable consequence. Over the $5$ random seeds of our sampling stability sweep (Appendix A.5), we examine the tail of the per-utterance WER distribution across all $5440$ generations per system (Table 7). The AR backbone collapses completely on 5 generations, abandoning the target text, while DELTA-TTS never does and its worst case remains a partial error at $64\%$. The error decomposition points the same way: insertions, the signature of repetition-style hallucination, drop by $47.9\%$ (Appendix A.6).

## 5 Conclusion

We presented DELTA-TTS, a lightweight framework that converts a pretrained autoregressive TTS model into a discrete diffusion language model with only about $15\%$ added parameters. On top of LoRA, a block-wise convolution module injected the local acoustic context that bidirectional attention captures weakly, and a $1/t$-weighted objective with a time-shifted schedule deferred low-confidence positions to later steps. Trained on only $585$ hours of LibriTTS, DELTA-TTS achieved a $1.75\%$ WER on Seed-TTS *test-en*, improving on the $2.02\%$ of its AR backbone while generating tokens $3.3\times$ faster. We attributed these gains to local mixing and to confidence-ordered decoding, which deferred the positions where the AR model is least confident until sufficient context has accumulated and therefore reduced hallucination and error propagation. These results support our hypothesis that confidence-ordered decoding produces fewer catastrophic failures than strictly left-to-right decoding.

## Acknowledgments and Disclosure of Funding

This work was partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2026-25602529). This work was also partly supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grants funded by the Korea government (MSIT) (IITP-2026-RS-2026-25615334, AI Star Fellowship Support Program; RS-2019-II190421, Artificial Intelligence Graduate School Program (Sungkyunkwan University)). This work was also partly supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP, AI Computing Support Project for R&D) grant funded by the Korea government (MSIT) (High-Performance Research AI Computing Infrastructure Support at the 2 PFLOPS Scale, 25%).

## References


[1] Philip Anastassiou, Jiawei Chen, Jitong Chen, Yuanzhe Chen, Zhuo Chen, Ziyi Chen, Jian Cong, Lelai Deng, Chuang Ding, Lu Gao, et al. Seed-TTS: A family of high-quality versatile speech generation models. *arXiv preprint arXiv:2406.02430*, 2024.

[2] Marianne Arriola, Aaron Gokaslan, Justin Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. In *International Conference on Learning Representations*, 2025.

[3] Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021.

[4] Zalán Borsos, Matt Sharifi, Damien Vincent, Eugene Kharitonov, Neil Zeghidour, and Marco Tagliasacchi. SoundStorm: Efficient parallel audio generation, 2024.

[5] Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. WavLM: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518, 2022.

[6] Sanyuan Chen, Chengyi Wang, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, et al. Neural codec language models are zero-shot text to speech synthesizers. *IEEE Transactions on Audio, Speech and Language Processing*, 33:705–718, 2025.

[7] Yushen Chen, Zhikang Niu, Ziyang Ma, Keqi Deng, Chunhui Wang, Jian Zhao, Kai Yu, and Xie Chen. F5-TTS: A fairytaler that fakes fluent and faithful speech with flow matching. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6255–6271, 2025.

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

[9] Sander Dieleman, Laurent Sartran, Arman Roshannai, Nikolay Savinov, Yaroslav Ganin, Pierre H Richemond, Arnaud Doucet, Robin Strudel, Chris Dyer, Conor Durkan, et al. Continuous diffusion for categorical data. *arXiv preprint arXiv:2211.15089*, 2022.

[10] Zhihao Du, Qian Chen, Shiliang Zhang, Kai Hu, Heng Lu, Yexin Yang, Hangrui Hu, Siqi Zheng, Yue Gu, Ziyang Ma, et al. CosyVoice: A scalable multilingual zero-shot text-to-speech synthesizer based on supervised semantic tokens. *arXiv preprint arXiv:2407.05407*, 2024.

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

[12] Zhihao Du, Yuxuan Wang, Qian Chen, Xian Shi, Xiang Lv, Tianyu Zhao, Zhifu Gao, Yexin Yang, Changfeng Gao, Hui Wang, et al. CosyVoice 2: Scalable streaming speech synthesis with large language models. *arXiv preprint arXiv:2412.10117*, 2024.

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

[14] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

[15] Shansan Gong, Shivam Agarwal, Yizhe Zhang, Jiacheng Ye, Lin Zheng, Mukai Li, Chenxin An, Peilin Zhao, Wei Bi, Jiawei Han, et al. Scaling diffusion language models via adaptation from autoregressive models. *arXiv preprint arXiv:2410.17891*, 2024.

[16] Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. DiffuSeq: Sequence to sequence text generation with diffusion models. In *The Eleventh International Conference on Learning Representations*, 2023.

[17] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. Conformer: Convolution-augmented Transformer for Speech Recognition. In *Interspeech 2020*, pages 5036–5040, 2020.

[18] Hao-Han Guo, Yao Hu, Fei-Yu Shen, Xu Tang, Yi-Chen Wu, Feng-Long Xie, and Kun Xie. FireRedTTS-1S: An upgraded streamable foundation text-to-speech system. *arXiv preprint arXiv:2503.20499*, 2025.

[19] Xiaochuang Han, Sachin Kumar, and Yulia Tsvetkov. SSD-LM: Semi-autoregressive simplex-based diffusion language model for text generation and modular control. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11575–11596, 2023.

[20] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020.

[21] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[22] Hangrui Hu, Xinfa Zhu, Ting He, Dake Guo, Bin Zhang, Xiong Wang, Zhifang Guo, Ziyue Jiang, Hongkun Hao, Zishan Guo, et al. Qwen3-TTS technical report. *arXiv preprint arXiv:2601.15621*, 2026.

[23] Dongya Jia, Zhuo Chen, Jiawei Chen, Chenpeng Du, Jian Wu, Jian Cong, Xiaobin Zhuang, Chumin Li, Zhen Wei, Yuping Wang, and Yuxuan Wang. DiTAR: Diffusion transformer autoregressive modeling for speech generation. In *Forty-second International Conference on Machine Learning*, 2025.

[24] Zeqian Ju, Yuancheng Wang, Kai Shen, Xu Tan, Detai Xin, Dongchao Yang, Eric Liu, Yichong Leng, Kaitao Song, Siliang Tang, Zhizheng Wu, Tao Qin, Xiangyang Li, Wei Ye, Shikun Zhang, Jiang Bian, Lei He, Jinyu Li, and Sheng Zhao. Naturalspeech 3: Zero-shot speech synthesis with factorized codec and diffusion models. In *Forty-first International Conference on Machine Learning*, 2024.

[25] Semin Kim, Seungjun Chung, Taehong Moon, Sangheon Lee, Minyoung Ahn, Keon Lee, Nam Soo Kim, Jaewoong Cho, Ludwig Schmidt, Kangwook Lee, and Dongmin Park. Raon-opentts: Open models and data for robust text-to-speech. *arXiv preprint arXiv:2605.20830*, 2026.

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

[27] Keon Lee, Dong Won Kim, Jaehyeon Kim, Seungjun Chung, and Jaewoong Cho. DiTTo-TTS: Diffusion transformers for scalable text-to-speech without domain-specific factors. In *The Thirteenth International Conference on Learning Representations*, 2025.

[28] Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-LM improves controllable text generation. *Advances in neural information processing systems*, 35:4328–4343, 2022.

[29] Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Forty-first International Conference on Machine Learning*, 2024.

[30] Aleksandr Meister, Matvei Novikov, Nikolay Karpov, Evelina Bakhturina, Vitaly Lavrukhin, and Boris Ginsburg. LibriSpeech-PC: Benchmark for evaluation of punctuation and capitalization capabilities of end-to-end ASR models. In *2023 IEEE automatic speech recognition and understanding workshop (ASRU)*, pages 1–7. IEEE, 2023.

[31] Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[32] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR, 2023.

[33] Takaaki Saeki, Detai Xin, Wataru Nakata, Tomoki Koriyama, Shinnosuke Takamichi, and Hiroshi Saruwatari. UTMOS: UTokyo-SaruLab system for VoiceMOS challenge 2022. *arXiv preprint arXiv:2204.02152*, 2022.

[34] Subham S Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

[35] Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. *Advances in neural information processing systems*, 37:103131–103167, 2024.

[36] Jaesung Tae, Hamish Ivison, Sachin Kumar, and Arman Cohan. TESS 2: A large-scale generalist diffusion language model. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 21171–21188, 2025.

[37] Xinsheng Wang, Mingqi Jiang, Ziyang Ma, Ziyu Zhang, Songxiang Liu, Linqin Li, Zheng Liang, Qixi Zheng, Rui Wang, Xiaoqin 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.

[38] Yuancheng Wang, Haoyue Zhan, Liwei Liu, Ruihong Zeng, Haotian Guo, Jiachen Zheng, Qiang Zhang, Xueyao Zhang, Shunsi Zhang, and Zhizheng Wu. MaskGCT: Zero-shot text-to-speech with masked generative codec transformer. In *The Thirteenth International Conference on Learning Representations*, 2025.

[39] Yuancheng Wang, Jiachen Zheng, Junan Zhang, Xueyao Zhang, Huan Liao, and Zhizheng Wu. Metis: A foundation speech generation model with masked generative pre-training. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[40] Chengyue Wu, Hao Zhang, Shuchen Xue, Shizhe Diao, Yonggan Fu, Zhijian Liu, Pavlo Molchanov, Ping Luo, Song Han, and Enze Xie. Fast-dLLM v2: Efficient block-diffusion LLM. In *The Fourteenth International Conference on Learning Representations*, 2026.

[41] Kun Xie, Feiyu Shen, Junjie Li, Fenglong Xie, Xu Tang, and Yao Hu. FireRedTTS-2: Towards long conversational speech generation for podcast and chatbot. *arXiv preprint arXiv:2509.02020*, 2025.

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

[43] Zhen Ye, Xinfa Zhu, Chi-Min Chan, Xinsheng Wang, Xu Tan, Jiahe Lei, Yi Peng, Haohe Liu, Yizhu Jin, Zheqi Dai, et al. Llasa: Scaling train-time and inference-time compute for LLaMA-based speech synthesis. *arXiv preprint arXiv:2502.04128*, 2025.

[44] Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J. Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu. LibriTTS: A Corpus Derived from LibriSpeech for Text-to-Speech. In *Interspeech 2019*, pages 1526–1530, 2019.

[45] Siyi Zhou, Yiquan Zhou, Yi He, Xun Zhou, Jinchao Wang, Wei Deng, and Jingchen Shu. 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, 2026.

[46] Yixuan Zhou, Guoyang Zeng, Xin Liu, Xiang Li, Renjie Yu, Ziyang Wang, Runchuan Ye, Weiyue Sun, Jiancheng Gui, Kehan Li, et al. VoxCPM: Tokenizer-free TTS for context-aware speech generation and true-to-life voice cloning. *arXiv preprint arXiv:2509.24650*, 2025.

[47] Zhanhui Zhou, Lingjie Chen, Hanghang Tong, and Dawn Song. dLLM: Simple diffusion language modeling. *arXiv preprint arXiv:2602.22661*, 2026.

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

# A Appendix

## A.1 Implementation Details

We train on a single NVIDIA A100 with bfloat16 mixed precision. The LoRA targets the query, key, value, output, gate, up, and down projections of every transformer block, with rank $r{=}64$ and scaling $\alpha{=}128$. The depth-wise convolution uses kernel size 31 and dropout $0.1$. We optimize with AdamW at a constant learning rate of $1{\times}10^{-4}$ after a 2000-step linear warmup, with an effective batch size of 16 utterances obtained via gradient accumulation.

## A.2 End-to-End Latency

Table 8: End-to-end RTF including the flow-matching vocoder, on Seed-TTS *test-en*.

| Batch | CosyVoice3 (AR) | DELTA-TTS (8 steps) | | DELTA-TTS (16 steps) | |
|---|---|---|---|---|---|
| | RTF ↓ | RTF ↓ | Speedup ↑ | RTF ↓ | Speedup ↑ |
| 1 | 0.5967 | 0.1453 | 4.11× | 0.2201 | 2.71× |
| 2 | 0.3403 | 0.0760 | 4.48× | 0.1081 | 3.15× |
| 4 | 0.2108 | 0.0421 | **5.01**× | 0.0612 | **3.44**× |
| 8 | 0.1343 | 0.0297 | 4.52× | 0.0401 | 3.35× |

We measure RTF over the full generation pipeline, including text processing, speech-token generation, and the flow-matching vocoder, so that the reported numbers reflect user-visible latency rather than the token-generation stage alone. All measurements use a single NVIDIA A100. CosyVoice3 runs its official decoder with KV caching enabled, and DELTA-TTS is measured at 8 and 16 unmasking steps. RTF is wall-clock time divided by generated-audio duration, averaged over Seed-TTS *test-en*.

The speedup holds across batch sizes and peaks at batch 4. It is smaller than the speedup measured on the token-generation stage alone at batch 1, since the shared flow-matching vocoder is unaffected by the conversion and takes a fixed share of the wall-clock time in both systems.

## A.3 Alignment Sharpness Control

Table 9: Alignment sharpness control on Seed-TTS *test-en*. Sharpness is the fraction of text-to-speech attention mass falling within $\pm1$ text token of the position identified by MFA forced alignment, taken from each utterance's best-aligned head.

| Model | WER (%) ↓ | SIM ↑ | UTMOS ↑ | Sharpness ↑ | Relative |
|---|---|---|---|---|---|
| CosyVoice3 (backbone) | 2.02 | **0.692** | 3.955 | 0.725 | – |
| + LoRA fine-tune (control) | 1.87 | 0.687 | **3.982** | 0.720 | $-0.7\%$ |
| DELTA-TTS | **1.75** | 0.688 | 3.975 | **0.788** | $+8.8\%$ |

The sharper alignment in Figure 3(a) could in principle come from fine-tuning on a narrow domain rather than from the conversion itself. We therefore train a control that keeps attention causal, using the identical LoRA configuration, the same LibriTTS data, and the official CosyVoice3 objective. We then measure diagonal alignment sharpness on Seed-TTS *test-en*.

Fine-tuning on LibriTTS improves the backbone's WER from $2.02\%$ to $1.87\%$. The control's attention nonetheless does not sharpen, while DELTA-TTS is $8.8\%$ sharper, indicating that the sharper alignment does not follow from training on this data.

## A.4 Expressivity Preservation

The adapter is trained on LibriTTS, a corpus of read audiobook speech with little emotional or rhythmic variation, so we test whether the conversion preserves the expressive capability of the multi-

Table 10: Emotion cloning on the CV3-eval emotion subset (en, 150 utterances) with happy, sad, and angry prompts. Spread is the gap between the extreme emotions; parentheses give DELTA-TTS's spread as a fraction of the backbone's.

| Metric | Model | sad | happy | angry | Spread |
|---|---|---|---|---|---|
| Median F0 (Hz) | CosyVoice3 | 162.2 | 182.3 | 205.7 | 43.5 |
| | DELTA-TTS | 162.6 | 172.7 | 200.6 | 38.1 (88%) |
| F0 variability (st) | CosyVoice3 | 4.38 | 4.57 | 5.04 | 0.67 |
| | DELTA-TTS | 4.54 | 4.71 | 5.24 | 0.69 (104%) |
| Speaking rate (char/s) | CosyVoice3 | 12.9 | 13.9 | 15.4 | 2.5 |
| | DELTA-TTS | 16.0 | 17.9 | 18.3 | 2.3 (92%) |

Table 11: Expressive continuation on the CV3-eval subjective-continuation subset (en, 90 utterances). Values are the correlation between the prompt's attribute and the generation's, with bootstrap 95% intervals.

| Axis (prompt → generation) | CosyVoice3 | DELTA-TTS |
|---|---|---|
| Speaking rate | $+0.94$ $[0.88, 0.98]$ | $+0.87$ $[0.76, 0.95]$ |
| Volume (RMS dB) | $+0.98$ $[0.96, 0.99]$ | $+0.93$ $[0.87, 0.97]$ |
| Emotion (F0 variability, st) | $+0.76$ $[0.63, 0.88]$ | $+0.55$ $[0.29, 0.73]$ |
| Rhythm (F0 variability, st) | $+0.76$ $[0.37, 0.92]$ | $+0.70$ $[0.25, 0.88]$ |
| Pooled (90, z-scored) | $+0.85$ $[0.78, 0.91]$ | $+0.74$ $[0.64, 0.82]$ |

lingual backbone. We evaluate on the two expressive subsets of CV3-eval, measuring CosyVoice3 and DELTA-TTS under the same pipeline.

DELTA-TTS preserves the emotion ordering on every metric (sad $<$ happy $<$ angry) and retains 88–104% of the backbone's separation between emotions. The speaking rate is uniformly faster, with total, voiced, and silence durations all about 80% of the backbone's, which we attribute to the rule-based length estimator (Section 4.4): the short prompts in CV3-eval make the target length harder to estimate.

The prompt's expressive attributes are carried into the generation rather than flattened, most strongly for speed and volume. The adapter was never exposed to expressive prompts of this kind during adaptation, so the emotional and prosodic conditioning measured in both tables appears to be inherited from the backbone rather than learned from the training data.

### A.5 Sampling Stability Sweep

Table 12: Sampling stability on Seed-TTS *test-en*. Stochastic results are mean±std over 5 random seeds.

| Configuration | WER (%) ↓ | SIM ↑ | UTMOS ↑ |
|---|---|---|---|
| DELTA-TTS, stochastic | $\mathbf{1.75 \pm 0.10}$ | $0.688 \pm 0.001$ | $\mathbf{3.975 \pm 0.005}$ |
| CosyVoice3, stochastic | $1.99 \pm 0.09$ | $\mathbf{0.692 \pm 0.001}$ | $3.955 \pm 0.004$ |

We evaluate sampling stability by running each model with 5 different random seeds on Seed-TTS *test-en* (1088 utterances) under the default configuration (DELTA-TTS: $\mu = 0.3$). Since Seed-TTS provides no dev split, $\mu$ was selected on a held-out LibriTTS dev-clean set rather than on the evaluation set. We set top_p=0.8 for both models, following CosyVoice3's official sampling configuration.[2]

[2] `https://github.com/FunAudioLLM/CosyVoice`

### A.6 Error-Type Decomposition

Table 13: WER decomposed into substitutions, insertions, and deletions on Seed-TTS *test-en*, as a percentage of reference words.

| Error type (% of reference words) | CosyVoice3 (AR) | DELTA-TTS | Relative change |
|---|---|---|---|
| Substitution | 1.559 | **1.313** | $-15.8\%$ |
| Insertion | 0.117 | **0.061** | $-47.9\%$ |
| Deletion | **0.315** | 0.351 | $+11.4\%$ |

We decompose the WER on Seed-TTS *test-en* into substitutions, insertions, and deletions. Insertions, the signature of repetition-style hallucination, drop by $47.9\%$ and substitutions by $15.8\%$, while deletions are marginally higher (Table 13).

### A.7 Subjective Evaluation Details

Table 14: A/B preference test on 100 utterances drawn independently of the MOS set.

| | CosyVoice3 | Neutral | DELTA-TTS |
|---|---|---|---|
| Overall quality | 21% | 43% | **36%** |
| Speaker similarity | 22% | 49% | **29%** |

For the subjective evaluation in Table 2, we recruited 30 adult volunteers anonymously from the authors' institution, each rating 100 samples per system from Seed-TTS *test-en*. Listeners rated SMOS (speaker similarity to the prompt voice, 1–5) for all three systems and CMOS (naturalness against ground truth, $-3$ to $+3$) for the two synthetic ones. DELTA-TTS obtains significantly higher CMOS and SMOS than CosyVoice3 (Wilcoxon signed-rank test; $p < 0.001$ and $p < 0.01$).

We additionally run an A/B preference test with the same raters on 100 utterances drawn independently of the MOS set. Excluding neutral choices, DELTA-TTS is preferred significantly more often on both axes (Table 14; binomial test; $p < 0.001$ for quality, $p < 0.01$ for speaker similarity).

In our SMOS setup, raters hear the reference prompt and rate all systems against it. Since the ground truth is a different utterance by the same speaker, it naturally differs from the reference in style and recording conditions, while the TTS systems explicitly target that reference. Synthetic samples scoring above the ground truth are therefore an expected outcome of this protocol.

### A.8 Time-Shifted Schedule and $1/t$ Loss

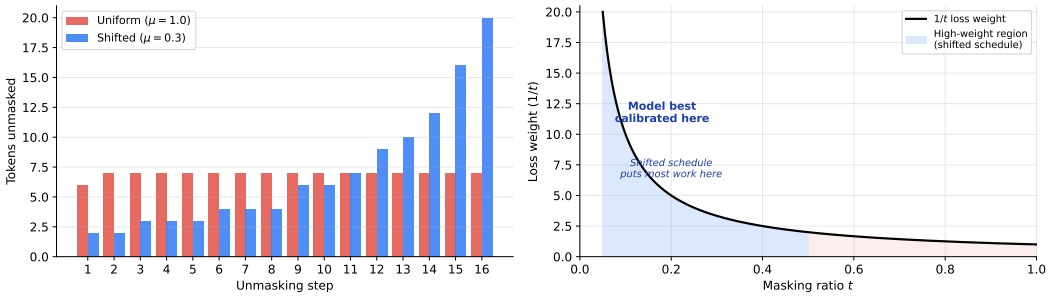



Figure 5: Number of tokens unmasked per step under uniform and shifted schedules (left), and the $1/t$ loss weight as a function of the masking ratio (right).

We use a time-shifted schedule that defers most unmasking decisions to the later steps of decoding. As shown in Figure 5, the shifted schedule concentrates these decisions at low masking ratios, where the $1/t$ loss places its largest weights.

### A.9 Easy-first, Hard-last Unmasking Order

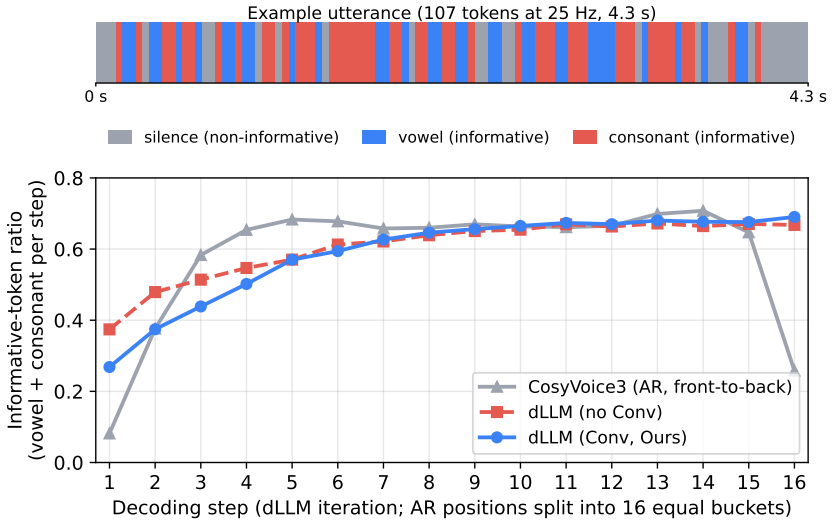



Figure 6: Per-step informative-token (vowel + consonant) ratio averaged over 1088 utterances. Strip on top: phoneme classes on one example utterance.

Figure 6 plots the per-step informative-token (vowel + consonant) ratio. Under the AR order (CosyVoice3 audio aligned via MFA), the ratio peaks at the middle steps and drops at both ends, since the AR decoder commits the leading and trailing silence at the first and last steps. Under DELTA-TTS, the ratio increases monotonically: the model commits silence first and defers phonetically meaningful tokens until enough bidirectional context has accumulated. This phoneme-class deferral mirrors the AR-difficulty deferral in Figure 7.

### A.10 Per-Step Commit Ordering

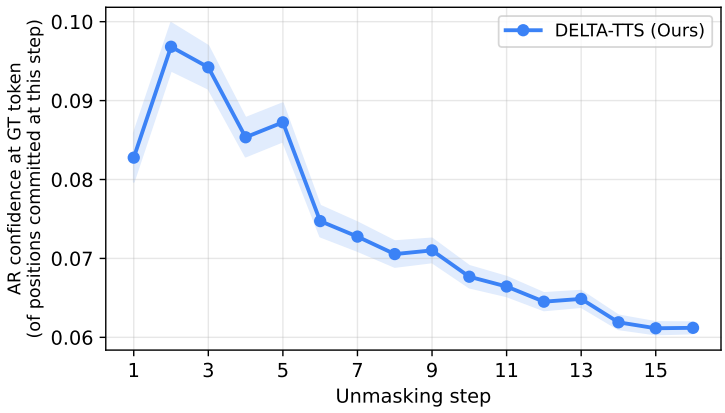



Figure 7: Mean teacher-forced AR confidence (GT token) of the positions DELTA-TTS commits at each unmasking step, over 1088 utterances. AR-low-confidence positions are deferred to later steps.

Figure 7 shows, at each unmasking step, the mean teacher-forced AR confidence of the positions DELTA-TTS commits. DELTA-TTS commits high-AR-confidence positions early and defers low-AR-confidence positions to later steps, once the bidirectional decoder has resolved more surrounding context. Through this deferral, DELTA-TTS commits even AR-hard positions at high confidence, producing the elevated distribution in Figure 4(b).

### A.11 Limitations and Future Work

We view DELTA-TTS as a starting point for converting AR TTS models into dLLMs, rather than a finished system. Specifically, the current setup determines the target length from a rule-based audio-to-character rate; this is a general challenge for dLLM-based generation, which maturing dLLM techniques may resolve. The model also operates only in English; multilingual extension remains future work. We also expect the conversion approach to transfer to other AR speech LMs and to richer multi-codebook tokenizers, and reward-based fine-tuning with metrics such as WER and speaker similarity is a natural next step.

### A.12 Broader Impacts

This work is intended for academic research purposes. DELTA-TTS shows that TTS can be obtained from a pretrained AR backbone with $585$ hours of public data and $\sim 15\%$ additional parameters, lowering the compute and data barrier for speech-synthesis research and bridging dLLM research between text and speech language modeling. As with any high-fidelity zero-shot TTS, the model carries misuse risks such as voice impersonation; we urge downstream users to obtain consent from any reference speaker and to clearly disclose synthetic audio.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction make three central claims: extending AR-to-dLLM conversion from text to speech (to our knowledge, the first); a Conformer-style convolution for short-range speech-token mixing; and $1.75\%$ WER on Seed-TTS test-en with a $3.3\times$ speedup over CosyVoice3 using only $585$ h of LibriTTS. All three are substantiated in Sections 3, 4.2, and 4.4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations in Appendix A.11: the rule-based target-length estimate (a general challenge for dLLM-based generation) and English-only training.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper presents an empirical study of AR-to-dLLM conversion for speech and does not include formal theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We describe the AR backbone (CosyVoice3), the LoRA and Conformer-style convolution adapters, the masked diffusion training objective with the shift operation, and the time-shifted unmasking schedule in Section 3. Implementation details (optimizer, learning rate, batch size, kernel size) are listed in Appendix A.1. We use the publicly released CosyVoice3 checkpoint, the public LibriTTS dataset, and the public Seed-TTS test-en and LibriSpeech-PC test-clean Subset B benchmarks; no proprietary data or model is involved.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: Code and trained model checkpoints will be released with full instructions for reproducing all experimental results. We include audio samples for the main systems in the supplementary material.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The training objective and unmasking schedule are described in Section 3 with explicit equations for the masked diffusion loss and the time-shifted schedule. Optimizer (AdamW, lr=$1\times10^{-4}$, 2,000-step linear warmup), batch size, depth-wise convolution kernel size, and dropout are listed in Appendix A.1. Inference hyperparameters (number of unmasking steps, $\mu$, top-$p$) are reported in the main text and in the sampling stability sweep (Appendix A.5).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Our main result on Seed-TTS *test-en* (1.75% WER) is the mean over 5 random seeds of stochastic decoding. The sampling stability sweep (Appendix A.5, Table 12) reports the corresponding mean±std for WER, SIM, and UTMOS for both DELTA-TTS and CosyVoice3 under matched sampling settings, demonstrating that DELTA-TTS's advantage holds within standard deviations.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix A.1 states that training is performed on a single NVIDIA A100 with bfloat16 mixed precision. Inference RTF is measured on a single NVIDIA A100 in Section 4.3.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have read and adhere to the NeurIPS Code of Ethics. The work uses only publicly available datasets (LibriTTS, Seed-TTS, LibriSpeech-PC) and a publicly released backbone (CosyVoice3). Our subjective evaluation involves volunteer human raters performing standard MOS listening tasks (audio similarity and naturalness ratings); this is a minimal-risk listening study with no sensitive personal data collected. Ethical considerations regarding zero-shot voice cloning are discussed in the Broader Impacts (Appendix A.12).

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Broader impacts are discussed in Appendix A.12: positive impacts include lowering compute and data barriers for speech-synthesis research and bridging dLLM research between text and speech language modeling; potential risks include misuse for voice impersonation, for which we encourage downstream consent and disclosure of synthetic audio.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: Code and trained model checkpoints will be released under a research-only license, with the usage guidance described in the Broader Impacts (Appendix A.12): downstream users are asked to obtain consent from any reference speaker and to clearly disclose synthetic audio.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite all assets used in the paper: the LibriTTS dataset [44], Seed-TTS *test-en* benchmark [1], LibriSpeech-PC *test-clean* Subset B (released by F5-TTS [7], derived from LibriSpeech), the CosyVoice3 backbone [11], and the evaluation models WavLM [5], ECAPA-TDNN [8], Whisper-Large-V3 [32], and UTMOS [33]. We use each asset within the terms of its license.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Audio samples in the supplementary material are accompanied by a brief README describing the systems compared and the sample selection criterion. Model adapter checkpoints will be released with documentation.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

Justification: Our subjective evaluation involves volunteer listeners who rate audio similarity (SMOS, 1–5) and comparative naturalness (CMOS, $-3$ to $+3$). The full rater-facing instructions are included in the supplementary material. Raters were unpaid volunteers, each rating 100 samples per system.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [Yes]

Justification: Our human evaluation involves only low-risk tasks (listening to short speech samples and rating style/quality). Participants were anonymous adult volunteers from the authors' institution and faced no physical or psychological risks.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs were not used as a component of our core methods. AI assistants were used only for minor editorial purposes such as grammar and spelling checks, which do not affect the scientific methodology, originality, or rigor of the research.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.