CUTETTS: EFFICIENT AND HIGH-QUALITY SPEECH SYNTHESIS VIA AUTOREGRESSIVE MODELING OF CONTINUOUS LATENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Zero-shot text-to-speech (TTS) now supports interactive assistants, personalized media, and accessibility tools. All TTS systems require faithful linguistic rendering, consistent speaker identity, and low-latency response. Yet compact streaming systems must preserve sufficient acoustic detail in a predictable low-rate latent sequence, while iterative diffusion sampling and classifier-free guidance multiply inference cost at every autoregressive step. To strike a balance between high-fidelity synthesis and low-latency inference, we present **CuteTTS**, a compact continuous-autoregressive TTS system. It combines semantically aligned causal VAE latents with patch-level autoregression, explicit speaker conditioning, and a bidirectional flow-matching head. We further introduce guidance–step distillation, which absorbs classifier-free guidance and multiple solver steps into a single interval-conditioned student. Evaluations on LibriSpeech and Seed-TTS-Eval demonstrate competitive intelligibility and speaker similarity in zero-shot voice cloning, while distillation lowers first-audio latency by 23.3% and real-time factor by 40.8% relative to the base model with comparable objective and subjective quality. These results provide a practical path toward continuous-autoregressive TTS that reconciles high-fidelity generation with the latency demands of real-time interaction.

1 INTRODUCTION

Zero-shot TTS enables conversational agents, personalized media, and assistive applications by reproducing the voice of an unseen speaker from a short reference utterance. Practical systems must preserve linguistic content and speaker identity while maintaining low latency. Existing approaches either model discrete codec tokens autoregressively (Wang et al., 2023a; Kharitonov et al., 2023) or generate continuous speech representations through diffusion or flow matching (Le et al., 2023; Chen et al., 2025). Hybrid streaming systems combine both paradigms (Du et al., 2024a;b). Despite substantial progress, achieving high-quality, speaker-faithful, and low-latency synthesis within a compact model remains challenging.

Continuous-autoregressive TTS retains the fidelity of continuous speech latents while supporting causal streaming generation (Peng et al., 2025; Rouard et al., 2025). Patch-based methods use a causal backbone to model dependencies across patches and a diffusion head to capture local correlations (Jia et al., 2025; Zhou et al., 2025b); recent variants also predict spectrogram frames directly or employ per-token diffusion, flow-based decoders, and consistency heads (Meng et al., 2025; Turetzky et al., 2024; Wang et al., 2025a; Wu et al., 2025; Zhou et al., 2026). Compact streaming systems nevertheless face two challenges: low-rate latents must balance reconstruction quality with autoregressive predictability, while iterative patch generation and LM-level classifier-free guidance require repeated model evaluations (Ho & Salimans, 2022). This overhead directly increases streaming latency.

In order to address these challenges, we present CuteTTS, a compact continuous-autoregressive TTS system built around a low-rate causal speech representation, patch-level language modeling, and a flow-matching-based diffusion head. The representation is trained with a semantic alignment objective, and a separate speaker embedding directly conditions the TTS backbone. To accelerate the diffusion head, we train a single student on interval-averaged velocities along the CFG-guided

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

teacher trajectory. The student therefore absorbs both the guidance branch and multiple numerical integration steps in one distillation objective.

Our main contributions are summarized as follows:

- We develop CuteTTS, an approximately 0.2B-parameter streaming system that combines patch-level autoregression over continuous speech latents with a flow-matching diffusion head. It achieves competitive intelligibility and speaker similarity across LibriSpeech and Seed-TTS benchmarks, and is preferred to similarly sized baselines in subjective sound-quality comparisons.
- We show that semantic alignment of the low-rate speech representation improves intelligibility, while explicit speaker conditioning strengthens zero-shot voice cloning. Additional controlled analyses characterize how representation rate, patch size, and module capacity trade off reconstruction fidelity, intelligibility, and acoustic quality.
- We introduce guidance–step distillation, which unifies classifier-free guidance removal and interval-wise integration in a single diffusion head. Under the paired efficiency protocol, the distilled model lowers mean first-audio latency by 23.3% and mean real-time factor by 40.8% while preserving synthesis quality close to that of the base model.

2 CuteTTS

Given text $\boldsymbol{y}$ and a reference utterance $\boldsymbol{x}^{\mathrm{ref}}$, CuteTTS autoregressively generates a sequence of continuous latent patches and decodes them into waveform chunks. As illustrated in Figure 1, the system consists of a causal audio VAE, a patch encoder, a causal language-model backbone, a diffusion head, and a speaker encoder.

2.1 Causal audio VAE with semantic alignment

We separately train a causal audio VAE that maps a 24-kHz waveform $\boldsymbol{x}$ to continuous latents at 12.5 Hz and reconstructs $\hat{\boldsymbol{x}}$. Following Kumar et al. (2023) and Zhou et al. (2025b), its encoder and decoder use a DAC-style fully convolutional architecture with stacked causal convolutions and residual blocks supporting incremental encoding and decoding.

In order to avoid posterior variance collapse in continuous autoregressive modeling, we adopt a σ-VAE (Sun et al., 2024; Peng et al., 2025). The encoder predicts a posterior mean $\boldsymbol{\mu} = E_\phi(\boldsymbol{x})$, while the scale is sampled from a prescribed distribution p_σ rather than estimated from the input:

$$\sigma \sim p_\sigma, \qquad \boldsymbol{z} = \boldsymbol{\mu} + \sigma\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{1}$$

Reconstruction fidelity alone does not ensure that a low-rate representation is well suited to text-conditioned generation. Inspired by Semantic-VAE (Niu et al., 2025), we align the latent sequence with a frozen WavLM teacher (Chen et al., 2022). The teacher features are projected to the VAE dimension and temporally aligned to the latent rate, and a cosine loss encourages the two representations to encode shared linguistic structure.

The VAE objective combines multi-resolution Mel-spectrogram reconstruction, adversarial and feature-matching losses from multi-period and multi-resolution discriminators, a lightly weighted KL regularizer, and semantic alignment:

$$\mathcal{L}_{\mathrm{VAE}} = \lambda_{\mathrm{mel}}\mathcal{L}_{\mathrm{mel}}(\boldsymbol{x}, \hat{\boldsymbol{x}}) + \lambda_{\mathrm{adv}}\mathcal{L}_{\mathrm{adv}}(\hat{\boldsymbol{x}}) + \lambda_{\mathrm{feat}}\mathcal{L}_{\mathrm{feat}}(\boldsymbol{x}, \hat{\boldsymbol{x}}) + \lambda_{\mathrm{KL}}\mathcal{L}_{\mathrm{KL}} + \lambda_{\mathrm{sem}}\mathcal{L}_{\mathrm{sem}}. \tag{2}$$

where $\mathcal{L}_{\mathrm{sem}}$ is the negative cosine similarity between the latent sequence and the aligned teacher features. After VAE training, only the causal encoder and decoder are retained; the TTS model operates entirely in the continuous VAE space.

2.2 Explicit speaker conditioning

We train a compact ECAPA-style speaker encoder (Desplanques et al., 2020) by distilling a frozen WavLM Large speaker-verification teacher (Chen et al., 2022). Given a reference utterance $\boldsymbol{x}^{\mathrm{ref}}$,

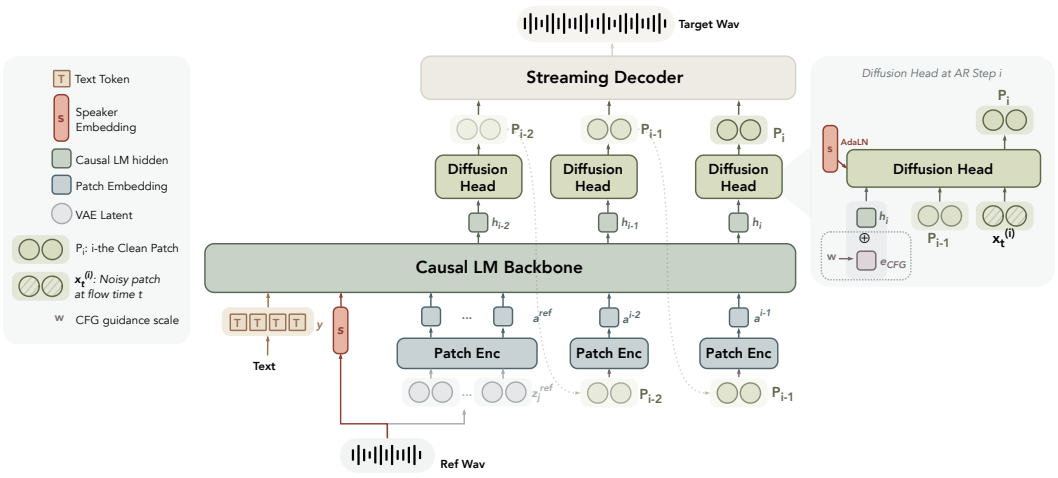


Figure 1: Overview of CuteTTS. A patch encoder aggregates previously generated latent patches for the causal backbone. The final-layer LM hidden state at the current autoregressive position, preceding latent patch, and speaker embedding condition the diffusion head to generate the next continuous patch; for the distilled model, CFG weight w is supplied as an additional condition. A causal VAE decoder converts completed patches to audio incrementally.

the student and teacher embeddings are $\boldsymbol{s}_{\text{student}} = E_{\text{student}}(\boldsymbol{x}^{\text{ref}})$ and $\boldsymbol{s}_{\text{teacher}} = E_{\text{teacher}}(\boldsymbol{x}^{\text{ref}})$, respectively. Both embeddings are L_2-normalized. The student is optimized with

$$\begin{aligned}\mathcal{L}_{\text{spk}} = \mathcal{L}_{\cos}(\boldsymbol{s}_{\text{student}}, \boldsymbol{s}_{\text{teacher}}) \\ + \lambda_{\text{pair}}\mathcal{L}_{\text{pair}}(\boldsymbol{s}_{\text{student}}, \boldsymbol{s}_{\text{teacher}}) \\ + \lambda_{\text{soft}}\mathcal{L}_{\text{soft}}(\boldsymbol{s}_{\text{student}}, \boldsymbol{s}_{\text{teacher}}),\end{aligned} \tag{3}$$

where $\mathcal{L}_{\cos}$ aligns each student embedding with its teacher target, $\mathcal{L}_{\text{pair}}$ matches pairwise speaker similarities within a minibatch, and $\mathcal{L}_{\text{soft}}$ matches the teacher-induced similarity distribution. Only the student encoder is retained for TTS, and we use $\boldsymbol{s} \equiv \boldsymbol{s}_{\text{student}}$ as the speaker condition. A learned projection inserts $\boldsymbol{s}$ into the causal backbone, while adaptive layer normalization conditions the diffusion head on the same embedding. Condition dropout provides the unconditional predictions used for classifier-free guidance.

2.3 Patch-based diffusion autoregression

Let $\boldsymbol{z}_0, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_{T-1}$ denote the target latent frames, and let K be the number of frames per patch. After padding the sequence to a multiple of K when necessary, we form N non-overlapping patches,

$$\boldsymbol{P}_i = \left(\boldsymbol{z}_{iK}, \boldsymbol{z}_{iK+1}, \ldots, \boldsymbol{z}_{(i+1)K-1}\right), \qquad i \in \{0, \ldots, N-1\}, \tag{4}$$

where N is the number of patches. CuteTTS uses $K = 2$, giving $\boldsymbol{P}_i = (\boldsymbol{z}_{2i}, \boldsymbol{z}_{2i+1})$. The patch encoder A is a bidirectional Transformer applied independently within each patch. It prepends a learned summary token to the latent frames, allows the token to attend to the entire patch, and uses its final state as the patch embedding $\boldsymbol{a}_i = A(\boldsymbol{P}_i)$. We partition the reference latent sequence in the same way, denoting its patches by $\boldsymbol{P}_j^{\text{ref}}$. The same encoder yields the reference conditioning sequence $\boldsymbol{a}^{\text{ref}} = (A(\boldsymbol{P}_0^{\text{ref}}), \ldots, A(\boldsymbol{P}_{M-1}^{\text{ref}}))$, where M is the number of reference patches. A causal Transformer summarizes the text, speaker condition, reference sequence, and previously generated patch embeddings at step i:

$$\boldsymbol{h}_i = F_\theta\left(\boldsymbol{y}, \boldsymbol{s}, \boldsymbol{a}^{\text{ref}}, \boldsymbol{a}_{<i}\right). \tag{5}$$

The diffusion head then models the next patch conditioned on $\boldsymbol{h}_i$, $\boldsymbol{s}$, and the preceding latent patch $\boldsymbol{P}_{i-1}$. This division of labor follows the patch-based formulation of DiTAR (Jia et al., 2025): the causal backbone captures inter-patch structure, while bidirectional attention within the diffusion head captures correlations inside a patch.

The diffusion head is trained with conditional flow matching (Lipman et al., 2023). For a clean patch $\boldsymbol{P}$ and Gaussian noise $\boldsymbol{\xi}$, we use the linear path

$$\boldsymbol{x}_t = (1-t)\boldsymbol{\xi} + t\boldsymbol{P}, \qquad \boldsymbol{v}^\star = \boldsymbol{P} - \boldsymbol{\xi}, \tag{6}$$

and regress the conditional velocity

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}\left[\|\boldsymbol{v}_\theta(\boldsymbol{x}_t, t \mid \boldsymbol{h}_i, \boldsymbol{s}, \boldsymbol{P}_{i-1}) - \boldsymbol{v}^\star\|_2^2\right]. \tag{7}$$

After sampling $\boldsymbol{P}_i$, the patch encoder produces $\boldsymbol{a}_i$ for the next autoregressive step. A stop head terminates generation, and the VAE decoder emits the corresponding waveform chunk.

2.4 Guidance–step distillation

Prior work either distills classifier-free guidance or reduces solver steps through progressive, consistency, and average-velocity objectives (Meng et al., 2023; Salimans & Ho, 2022; Song et al., 2023; Geng et al., 2025; Frans et al., 2025; Wang et al., 2025b). We combine both goals in one interval-conditioned head.

We use LM-level classifier-free guidance (Ho & Salimans, 2022) to construct the teacher trajectory. For the conditional branch, let $c_i = (\boldsymbol{h}_i, \boldsymbol{s}, \boldsymbol{P}_{i-1})$ denote the patch condition, and let $\boldsymbol{v}_T(\boldsymbol{x}, t \mid c_i)$ be the corresponding teacher velocity. The unconditional branch removes the text, reference, and speaker conditions while retaining the generated patch history:

$$\boldsymbol{h}_i^{\mathrm{u}} = F_\theta(\boldsymbol{a}_{<i}), \qquad c_i^{\mathrm{u}} = (\boldsymbol{h}_i^{\mathrm{u}}, \boldsymbol{P}_{i-1}). \tag{8}$$

The CFG-guided teacher velocity is then

$$\boldsymbol{v}_T^{\mathrm{cfg}}(\boldsymbol{x}, t \mid c_i) = \boldsymbol{v}_T(\boldsymbol{x}, t \mid c_i) + w\left[\boldsymbol{v}_T(\boldsymbol{x}, t \mid c_i) - \boldsymbol{v}_T(\boldsymbol{x}, t \mid c_i^{\mathrm{u}})\right], \tag{9}$$

where w is the guidance scale. Computing this velocity requires one conditional and one unconditional forward pass at each solver step.

We distill the resulting teacher trajectory directly. Starting from a noised patch $\boldsymbol{x}_t$ defined as in Equation 6, let $\boldsymbol{x}_{t+\Delta t}^{\mathrm{teacher}}$ denote the endpoint obtained by numerically integrating the CFG-guided teacher velocity over $[t, t+\Delta t]$, where $\Delta t \in \mathcal{D}$ and $\mathcal{D}$ denotes the supported interval lengths. The student is conditioned on the starting time, interval length, and guidance scale, and predicts the teacher's average velocity over the interval,

$$\bar{\boldsymbol{v}}_T = \frac{\boldsymbol{x}_{t+\Delta t}^{\mathrm{teacher}} - \boldsymbol{x}_t}{\Delta t}, \qquad \mathcal{L}_{\mathrm{GS}} = \mathbb{E}\left[\|\boldsymbol{v}_S(\boldsymbol{x}_t, t, \Delta t \mid c_i, w) - \bar{\boldsymbol{v}}_T\|_2^2\right]. \tag{10}$$

At inference, a single student forward pass maps $\boldsymbol{x}_t$ directly to $\boldsymbol{x}_{t+\Delta t}$,

$$\boldsymbol{x}_{t+\Delta t} = \boldsymbol{x}_t + \Delta t\, \boldsymbol{v}_S(\boldsymbol{x}_t, t, \Delta t \mid c_i, w). \tag{11}$$

The target in Equation 10 is already CFG-guided, so the student does not require a separate unconditional branch. Because the interval is an input, the same diffusion head can be evaluated with different step budgets. In this sense, guidance removal and step reduction are learned jointly rather than as two sequential distillation stages.

3 Experiments

3.1 Experimental Setup

Implementation details. The TTS backbone contains approximately 230M parameters and comprises a bidirectional patch encoder, a causal language-model backbone, and a bidirectional diffusion head. The audio VAE encodes 24-kHz speech into 64-dimensional latents at 12.5 Hz. We group two frames per patch, resulting in an LM token rate of 6.25 tokens per second.

For each generated patch, the base system uses 10 diffusion-head function evaluations (NFEs) on each of the two CFG branches, whereas the student uses 4 NFEs on a single branch. Further training and inference details are provided in Appendix A.

Table 1: Objective evaluation results of zero-shot voice-cloning performance. "Disc." and "Cont." denote discrete and continuous speech representations; "AR" and "NAR" denote autoregressive and non-autoregressive generation. ↑ and ↓ indicate that higher and lower values are better, respectively. A dash denotes an unavailable result.

Type	Model	Params.	LibriSpeech test-clean		Seed-TTS EN		Seed-TTS ZH	
			WER (%)↓	SIM↑	WER (%)↓	SIM↑	WER (%)↓	SIM↑
Disc. AR	MOSS-TTS	8B	1.98	67.7	1.84	70.9	1.37	77.0
	Qwen3-TTS	1.7B	2.35	70.3	1.66	71.4	**0.91**	77.0
	FireRedTTS-2	1.5B	4.32	64.2	1.95	66.5	1.14	73.6
	MOSS-TTS-Nano	0.1B	4.10	48.4	4.62	49.9	3.13	64.3
Cont. NAR	F5-TTS	0.3B	2.42	66.0	1.83	67.0	1.56	76.0
	ZipVoice	0.1B	2.05	67.4	1.70	69.7	1.40	75.1
Disc. AR + Cont. NAR	IndexTTS2	1.5B	2.47	70.0	2.22	70.6	1.02	76.5
	CosyVoice 3	0.5B	1.99	69.7	2.02	71.8	1.16	78.0
Cont. AR	VoxCPM2	2B	3.01	74.0	1.84	75.3	0.97	**79.5**
	VibeVoice	1.5B	–	–	3.04	68.9	1.16	74.4
	DiTAR	0.6B	2.39	67.0	1.69	73.5	1.02	75.3
	VibeVoice-Realtime	0.5B	2.00	69.5	2.05	63.3	–	–
	Pocket TTS	0.1B	**1.59**	49.1	**1.63**	50.7	–	–
Cont. AR	**CuteTTS**	0.2B	2.16	**78.9**	2.04	**76.5**	1.41	77.8
	CuteTTS-distill	0.2B	2.41	76.8	2.03	74.2	1.47	75.6

Training and evaluation datasets. CuteTTS, including its audio VAE and speaker encoder, is trained on an internal multilingual dataset containing approximately 550,000 hours of speech. For objective evaluation, we use LibriSpeech test-clean (Panayotov et al., 2015) under the protocol adopted by F5-TTS (Chen et al., 2025), together with the English and Chinese test sets released with Seed-TTS (Anastassiou et al., 2024). For subjective evaluation and efficiency benchmarking, we sample 50 examples from LibriSpeech test-clean and refer to this fixed set as the LibriSpeech test-clean subset.

Evaluation metrics. For objective evaluation, we compute Word Error Rate (WER) using Faster-Whisper-large-v3 for LibriSpeech test-clean, Whisper-large-v3 for Seed-TTS EN, and Paraformer-zh for Seed-TTS ZH (Radford et al., 2023; Gao et al., 2022). Speaker similarity (SIM) is measured as the cosine similarity between embeddings from a fine-tuned WavLM-Large model. For the ablation analyses, we additionally use UTMOS (Saeki et al., 2022) as an automatic estimate of perceived speech quality. For subjective evaluation, annotators compare paired outputs from two systems along two dimensions: sound quality and naturalness.

Baselines. We compare CuteTTS against a broad range of recent publicly available TTS systems, including discrete-autoregressive systems such as the MOSS-TTS family (Gong et al., 2026), Qwen3-TTS (Hu et al., 2026), and FireRedTTS-2 (Xie et al., 2025); continuous-non-autoregressive systems such as F5-TTS (Chen et al., 2025) and ZipVoice (Zhu et al., 2025); hybrid systems such as IndexTTS2 (Zhou et al., 2025a) and CosyVoice 3 (Du et al., 2025); and continuous-autoregressive systems such as VoxCPM2 (Zhou et al., 2026), the VibeVoice family (Peng et al., 2025; Microsoft, 2025), DiTAR (Jia et al., 2025), and Pocket TTS (Kyutai, 2026). All baseline results are obtained under the same evaluation protocol and evaluators described above. We use results from the original papers when they follow exactly the same protocol; otherwise, we re-evaluate the corresponding model outputs using our unified evaluation pipeline.

3.2 Experimental Results

Objective evaluation. Table 1 compares CuteTTS with publicly available systems across three zero-shot voice-cloning benchmarks. CuteTTS achieves performance comparable to leading systems with only 0.2B parameters, while maintaining low recognition error and strong speaker similarity across all three test sets. The distilled student closely matches the base model across the evaluated metrics while using a substantially smaller inference budget.

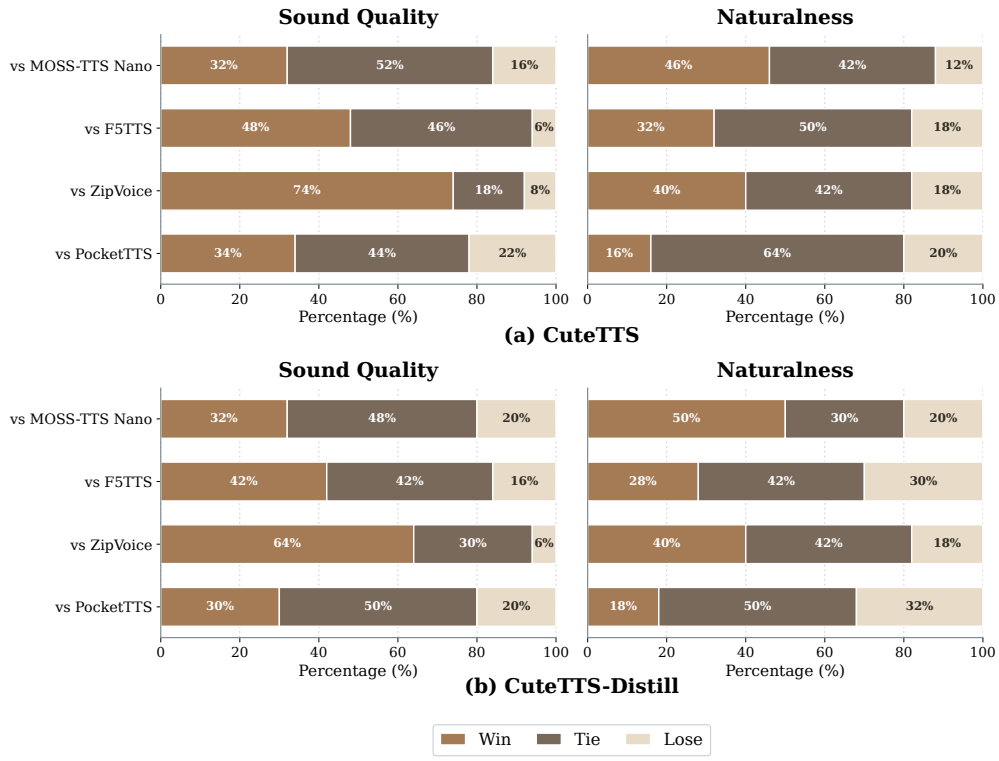


Figure 2: Subjective evaluation on the LibriSpeech test-clean subset. Results are reported as win/tie/loss percentages from the perspective of each CuteTTS variant.

Subjective evaluation. On the LibriSpeech test-clean subset, we compare CuteTTS and CuteTTS-distill with four similarly sized TTS systems: MOSS-TTS-Nano, F5-TTS, ZipVoice, and Pocket TTS. Annotators evaluate paired outputs independently in terms of sound quality and naturalness under a blinded protocol. Further details are provided in Appendix C.

Figure 2 reports win, tie, and loss percentages from the perspective of each CuteTTS variant. Both variants are preferred over all four baselines in terms of sound quality. The base model also receives more naturalness wins than losses against MOSS-TTS-Nano, F5-TTS, and ZipVoice, while its comparison with Pocket TTS is dominated by ties. The distilled model yields results closely matching those of the base model across both evaluation dimensions.

3.3 Efficiency

We define first-audio latency as the elapsed time from raw target text and a reference-audio path to the first PCM chunk on the CPU, and real-time factor (RTF) as total synthesis time divided by generated-audio duration. We report the mean, P50, and P95 first-audio latency, where P50 is the median and P95 is the latency threshold below which 95% of requests fall, characterizing tail latency. All efficiency measurements are conducted on an NVIDIA RTX 4090. Further benchmark details are provided in Appendix D.

Table 2 compares end-to-end inference efficiency across representative system families. For CuteTTS, guidance–step distillation reduces mean first-audio latency by 23.3% and mean RTF by 40.8% under this paired protocol. It also lowers P50 latency from 48.7 to 37.1 ms and P95 latency from 52.3 to 40.8 ms, showing consistent gains in both typical and tail latency.

Table 2: End-to-end efficiency across representative TTS systems, measured by first-audio latency and real-time factor (RTF).

Type	Model	Params.	Latency (ms)↓			RTF↓
			Mean	P50	P95	
Disc. AR	MOSS-TTS (vLLM-Omni)	8B	–	–	–	0.498
	Qwen3-TTS (vLLM-Omni)	1.7B	108.8	108.4	112.2	0.190
	FireRedTTS-2	1.5B	316.0	314.7	326.6	1.113
	MOSS-TTS-Nano	0.1B	123.8	122.7	142.8	0.778
Cont. NAR	F5-TTS	0.3B	628.2	622.2	667.5	0.122
	ZipVoice	0.1B	200.5	191.3	250.0	**0.030**
Disc. AR + Cont. NAR	IndexTTS2	1.5B	3826.1	3694.8	5490.0	0.676
	CosyVoice 3	0.5B	2370.1	2263.7	3345.4	0.453
Cont. AR	VoxCPM2	2B	66.6	63.0	82.2	0.404
	VibeVoice-Realtime	0.5B	62.9	62.6	64.8	0.365
	Pocket TTS	0.1B	51.0	50.7	52.9	0.125
Cont. AR	**CuteTTS**	0.2B	49.0	48.7	52.3	0.184
	CuteTTS-distill	0.2B	**37.6**	**37.1**	**40.8**	0.109

Table 3: Effect of semantic alignment on VAE reconstruction at a frame rate of 12.5 Hz on LibriSpeech test-clean.

Model	Rate (Hz)	SIM↑	PESQ-WB↑	STOI↑	UTMOS↑
w/ semantic	12.5	0.881	2.844	0.944	3.783
w/o semantic	12.5	0.881	2.885	0.945	3.786

4 Ablation Studies and Analysis

All ablation experiments use an approximately 50,000-hour subset of the full training corpus. Detailed experimental configurations are provided in Appendix A.4.

4.1 Temporal Granularity and Capacity Allocation

We denote a TTS model using a VAE frame rate of r Hz and p latent frames per patch as $\mathrm{TTS}_{r,p}$; its LM token rate is $q = r/p$. We train VAEs at $r \in \{6.25, 12.5, 25, 50\}$ Hz and study TTS variants in two controlled settings. The first fixes $p = 1$ and varies r, while the second fixes $q = 6.25$ tokens per second using $\mathrm{TTS}_{6.25,1}$, $\mathrm{TTS}_{12.5,2}$, $\mathrm{TTS}_{25,4}$, and $\mathrm{TTS}_{50,8}$. We additionally scale the causal backbone and diffusion head independently under fine- and coarse-grained autoregressive regimes. Full configurations and results are provided in Appendix B.

Higher-rate VAEs improve reconstruction, but, when patch size is fixed, they also lengthen the autoregressive sequence and make the TTS modeling task more difficult. At a fixed token rate, $\mathrm{TTS}_{12.5,2}$ provides a favorable balance between intelligibility, speaker similarity, and predicted perceptual quality. Module-wise scaling further indicates that causal-backbone capacity primarily benefits intelligibility, whereas diffusion-head capacity becomes more valuable for speaker fidelity and perceptual quality as patch size increases.

4.2 Effect of Semantic Alignment

To assess the effect of semantic alignment, we train an additional VAE using the same configuration as the 12.5-Hz VAE in Appendix B.1, but without the semantic-alignment objective. Table 3 shows that adding semantic alignment incurs virtually no degradation in reconstruction quality.

We then train a TTS model on the representation without semantic alignment using the same configuration as $\mathrm{TTS}_{12.5,2}$. Figure 3 compares the two models across CFG weights. The semantically aligned

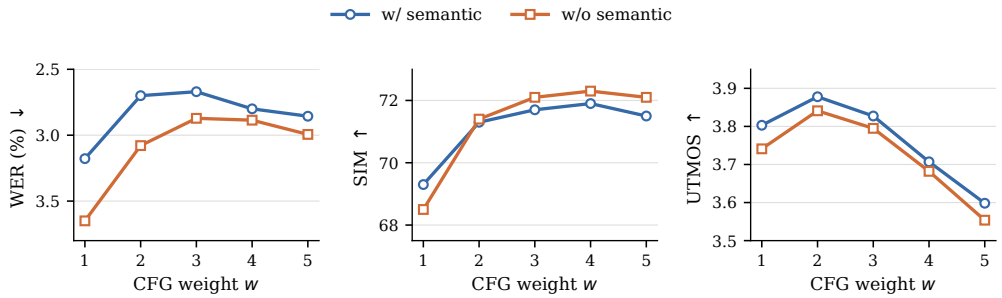


Figure 3: Effect of semantic alignment on TTS evaluation metrics across CFG weights on LibriSpeech test-clean. The WER axis is reversed so that better performance is shown higher.

Table 4: Effect of explicit speaker conditioning on $\mathrm{TTS}_{12.5,2}$ performance at CFG weight $w = 2$ on LibriSpeech test-clean.

Model	WER (%)↓	SIM↑	SIM (WeSpeaker)↑
$\mathrm{TTS}_{12.5,2}$	2.70	**71.3**	**82.5**
no spk	2.76	53.4	74.5
diff. head spk	**2.52**	71.2	**82.5**
spk as global token	2.76	71.0	82.4

representation substantially reduces WER across guidance strengths while retaining comparable SIM and UTMOS. These results indicate that semantic alignment makes the continuous representation easier for the causal backbone to model, improving intelligibility without compromising speaker similarity or predicted synthesis quality.

4.3 Effect of Explicit Speaker Conditioning

The default $\mathrm{TTS}_{12.5,2}$ uses the explicit speaker embedding in both the LM prefix and the diffusion head, with adaptive layer normalization (AdaLN) providing the diffusion-head condition. To assess the contribution of explicit speaker conditioning, we train two variants using the same training configuration: $\mathrm{TTS}_{12.5,2}$ (no spk), which removes the explicit speaker embedding from both components, and $\mathrm{TTS}_{12.5,2}$ (diff. head spk), which provides it only to the diffusion head. To examine the injection mechanism, we also train $\mathrm{TTS}_{12.5,2}$ (spk as global token), which retains LM-prefix speaker conditioning but injects the speaker embedding into the diffusion head as a global token rather than through AdaLN.

We compare all variants on LibriSpeech test-clean with CFG weight $w = 2$. Because our speaker encoder is distilled from WavLM and the primary SIM evaluator is also WavLM-based, proximity between their representation spaces could influence the measured similarity. We therefore additionally report SIM using WeSpeaker (Wang et al., 2023b) as an independent speaker-embedding model.

As shown in Table 4, removing the explicit speaker embedding substantially degrades similarity under both speaker evaluators, confirming its importance for zero-shot voice cloning. Conditioning the diffusion head alone recovers most of this loss, and using the speaker embedding as a global token instead of AdaLN provides no consistent benefit. The default $\mathrm{TTS}_{12.5,2}$ achieves the highest WavLM-based SIM, ties the best WeSpeaker SIM, and remains competitive in WER. It therefore provides the strongest overall balance and is retained as our default conditioning design.

4.4 Effect of Guidance–Step Distillation

To assess guidance–step distillation across inference budgets, we compare CuteTTS and CuteTTS-distill using 4, 2, and 1 diffusion-head NFEs and sweep the CFG weight from 1 to 5 on LibriSpeech test-clean. The base model uses standard two-branch LM-CFG with a Sway coefficient of -0.8,

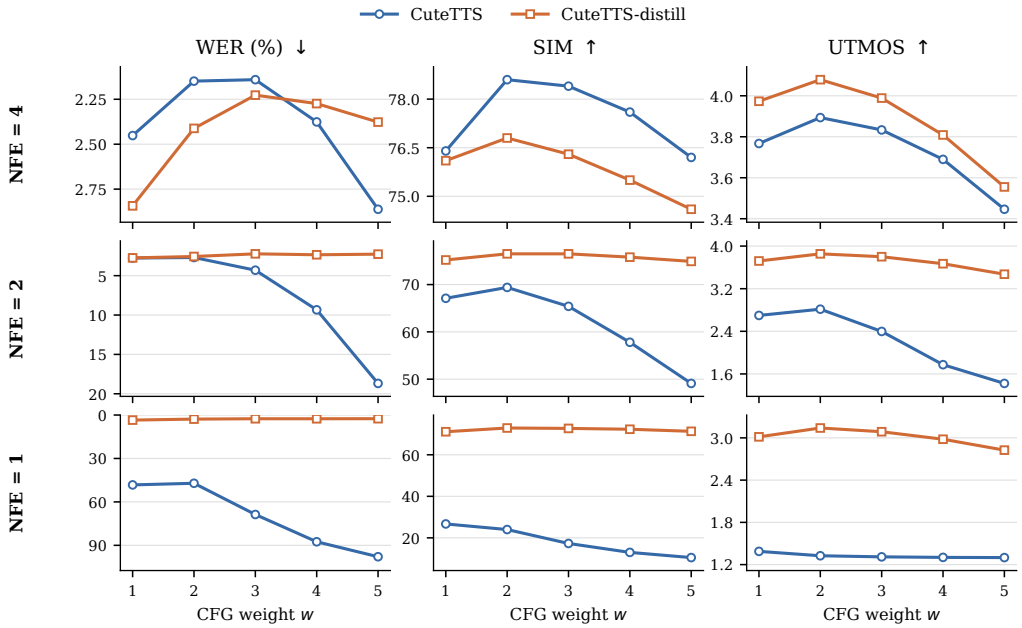


Figure 4: Effect of guidance–step distillation on TTS performance across CFG weights and diffusion-head NFE budgets on LibriSpeech test-clean. WER axes are reversed so that better performance is shown higher.

whereas the distilled model uses its single-branch distilled guidance and a uniform time grid. Figure 4 reports WER, SIM, and UTMOS for all settings.

At four NFEs, both models maintain strong performance over moderate guidance weights. As the NFE budget decreases, the base model degrades sharply, while the distilled model retains low WER and high speaker similarity at both two and one NFE. The distilled model always achieves higher UTMOS than the base model across the tested NFE budgets and CFG weights. These results show that guidance–step distillation substantially improves robustness under highly constrained sampling budgets.

5 Conclusion

We presented CuteTTS, a compact streaming TTS system that combines patch-level autoregression over continuous speech latents with a bidirectional flow-matching head. Its semantically aligned causal σ-VAE preserves acoustic detail at a low rate, while explicit speaker conditioning supplies a direct identity cue for zero-shot voice cloning. Across LibriSpeech and Seed-TTS benchmarks, CuteTTS achieves competitive intelligibility and speaker similarity, with paired subjective evaluations showing a consistent sound-quality preference over similarly sized baselines.

Our controlled studies reveal that representation rate and generation granularity must be considered jointly. Higher-rate VAEs improve reconstruction, but longer autoregressive sequences make TTS modeling more difficult; at a fixed token rate, moderate patching provides a favorable balance. Module-wise scaling indicates that causal-backbone capacity primarily benefits intelligibility, whereas diffusion-head capacity becomes more valuable for speaker fidelity and perceptual quality as patch size increases. Semantic alignment further improves intelligibility, and explicit speaker conditioning preserves identity. Finally, guidance–step distillation compresses two-branch CFG and multiple integration steps into a single head that remains effective at small NFE budgets. It reduces mean first-audio latency by 23.3% and mean real-time factor by 40.8% while maintaining synthesis quality close to the base model.

6 AI USE STATEMENT

Generative AI tools were used to assist with interpreting the reported experimental results. They were also used to summarize and analyze existing literature; support brainstorming and information retrieval; identify relevant literature; edit the manuscript for readability; format references; and suggest the manuscript structure. They were not used for other research tasks requiring disclosure under the ICLR 2027 AI Policy. All AI-assisted content and interpretations were independently reviewed and verified by the authors. The authors take responsibility for the final content of this work.

REFERENCES

Philip Anastassiou, Jiawei Chen, Jitong Chen, Yuanzhe Chen, Zhuo Chen, Ziyi Chen, Jian Cong, Lelai Deng, Chuang Ding, Lu Gao, Mingqing Gong, Peisong Huang, Qingqing Huang, Zhiying Huang, Yuanyuan Huo, Dongya Jia, Chumin Li, Feiya Li, Hui Li, Jiaxin Li, Xiaoyang Li, Xingxing Li, Lin Liu, Shouda Liu, Sichao Liu, Xudong Liu, Yuchen Liu, Zhengxi Liu, Lu Lu, Junjie Pan, Xin Wang, Yuping Wang, Yuxuan Wang, Zhen Wei, Jian Wu, Chao Yao, Yifeng Yang, Yuanhao Yi, Junteng Zhang, Qidi Zhang, Shuo Zhang, Wenjie Zhang, Yang Zhang, Zilin Zhao, Dejian Zhong, and Xiaobin Zhuang. Seed-TTS: A family of high-quality versatile speech generation models. *arXiv preprint arXiv:2406.02430*, 2024. URL https://arxiv.org/abs/2406.02430.

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. 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. doi: 10.1109/JSTSP.2022.3188113. URL https://doi.org/10.1109/JSTSP.2022.3188113.

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)*, pp. 6255–6271, 2025. doi: 10.18653/v1/2025.acl-long.313. URL https://aclanthology.org/2025.acl-long.313/.

Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN-based speaker verification. In *Interspeech 2020*, pp. 3830–3834, 2020. doi: 10.21437/Interspeech.2020-2650. URL https://www.isca-archive.org/interspeech_2020/desplanques20_interspeech.html.

Zhihao Du, Qian Chen, Shiliang Zhang, Kai Hu, Heng Lu, Yexin Yang, Hangrui Hu, Siqi Zheng, Yue Gu, Ziyang Ma, Zhifu Gao, and Zhijie Yan. Cosyvoice: A scalable multilingual zero-shot text-to-speech synthesizer based on supervised semantic tokens. *arXiv preprint arXiv:2407.05407*, 2024a. URL https://arxiv.org/abs/2407.05407.

Zhihao Du, Yuxuan Wang, Qian Chen, Xian Shi, Xiang Lv, Tianyu Zhao, Zhifu Gao, Yexin Yang, Changfeng Gao, Hui Wang, Fan Yu, Huadai Liu, Zhengyan Sheng, Yue Gu, Chong Deng, Wen Wang, Shiliang Zhang, Zhijie Yan, and Jingren Zhou. Cosyvoice 2: Scalable streaming speech synthesis with large language models. *arXiv preprint arXiv:2412.10117*, 2024b. URL https://arxiv.org/abs/2412.10117.

Zhihao Du, Changfeng Gao, Yuxuan Wang, Fan Yu, Tianyu Zhao, Hao Wang, Xiang Lv, Hui Wang, Chongjia Ni, Xian Shi, Keyu An, Guanrou Yang, Yabin Li, Yanni Chen, Zhifu Gao, Qian Chen, Yue Gu, Mengzhe Chen, Yafeng Chen, Shiliang Zhang, Wen Wang, and Jieping Ye. Cosyvoice 3: Towards in-the-wild speech generation via scaling-up and post-training. *arXiv preprint arXiv:2505.17589*, 2025. URL https://arxiv.org/abs/2505.17589.

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/559a0998fab1d19b80e7e43a5852401c-Abstract-Conference.html.

Zhifu Gao, Shiliang Zhang, Ian McLoughlin, and Zhijie Yan. Paraformer: Fast and accurate parallel transformer for non-autoregressive end-to-end speech recognition. In *Interspeech 2022*, pp. 2063–2067, 2022. doi: 10.21437/Interspeech.2022-9996. URL https://www.isca-archive.org/interspeech_2022/gao22b_interspeech.html.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *arXiv preprint arXiv:2505.13447*, 2025. URL https://arxiv.org/abs/2505.13447.

Yitian Gong, Botian Jiang, Yiwei Zhao, Yucheng Yuan, Kuangwei Chen, Yaozhou Jiang, Cheng Chang, Dong Hong, Mingshu Chen, Ruixiao Li, Yiyang Zhang, Yang Gao, Hanfu Chen, Ke Chen, Songlin Wang, Xiaogui Yang, Yuqian Zhang, Kexin Huang, ZhengYuan Lin, Kang Yu, Ziqi Chen, Jin Wang, Zhaoye Fei, Qinyuan Cheng, Shimin Li, and Xipeng Qiu. MOSS-TTS technical report. *arXiv preprint arXiv:2603.18090*, 2026. URL https://arxiv.org/abs/2603.18090.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. URL https://arxiv.org/abs/2207.12598.

Hangrui Hu, Xinfa Zhu, Ting He, Dake Guo, Bin Zhang, Xiong Wang, Zhifang Guo, Ziyue Jiang, Hongkun Hao, Zishan Guo, Xinyu Zhang, Pei Zhang, Baosong Yang, Jin Xu, Jingren Zhou, and Junyang Lin. Qwen3-TTS technical report. *arXiv preprint arXiv:2601.15621*, 2026. URL https://arxiv.org/abs/2601.15621.

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 *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 27255–27270. PMLR, 2025. URL https://proceedings.mlr.press/v267/jia25a.html.

Eugene Kharitonov, Damien Vincent, Zalán Borsos, Raphaël Marinier, Sertan Girgin, Olivier Pietquin, Matt Sharifi, Marco Tagliasacchi, and Neil Zeghidour. Speak, read and prompt: High-fidelity text-to-speech with minimal supervision. *Transactions of the Association for Computational Linguistics*, 11:1703–1718, 2023. doi: 10.1162/tacl_a_00618. URL https://aclanthology.org/2023.tacl-1.95/.

Rithesh Kumar, Prem Seetharaman, Alejandro Luebs, Ishaan Kumar, and Kundan Kumar. High-fidelity audio compression with improved RVQGAN. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2306.06546.

Kyutai. Pocket TTS: A high-quality TTS with voice cloning that runs on CPU. Kyutai technical report, 2026. URL https://kyutai.org/blog/2026-01-13-pocket-tts/.

Matthew Le, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, Rashel Moritz, Mary Williamson, Vimal Manohar, Yossi Adi, Jay Mahadeokar, and Wei-Ning Hsu. Voicebox: Text-guided multilingual universal speech generation at scale. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/2d8911db9ecedf866015091b28946e15-Abstract-Conference.html.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik P. Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14297–14306, 2023. doi: 10.1109/CVPR52729.2023.01374. URL https://openaccess.thecvf.com/content/CVPR2023/html/Meng_On_Distillation_of_Guided_Diffusion_Models_CVPR_2023_paper.html.

Lingwei Meng, Long Zhou, Shujie Liu, Sanyuan Chen, Bing Han, Shujie Hu, Yanqing Liu, Jinyu Li, Sheng Zhao, Xixin Wu, Helen M. Meng, and Furu Wei. Autoregressive speech synthesis

without vector quantization. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1287–1300, 2025. doi: 10.18653/v1/2025.acl-long.65. URL https://aclanthology.org/2025.acl-long.65/.

Microsoft. VibeVoice-Realtime-0.5B model card. Official Microsoft Hugging Face model card, 2025. URL https://huggingface.co/microsoft/VibeVoice-Realtime-0.5B.

Zhikang Niu, Shujie Hu, Jeongsoo Choi, Yushen Chen, Peining Chen, Pengcheng Zhu, Yunting Yang, Bowen Zhang, Jian Zhao, Chunhui Wang, and Xie Chen. Semantic-VAE: Semantic-alignment latent representation for better speech synthesis. *arXiv preprint arXiv:2509.22167*, 2025. URL https://arxiv.org/abs/2509.22167.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. LibriSpeech: An ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5206–5210, 2015. doi: 10.1109/ICASSP.2015.7178964. URL https://doi.org/10.1109/ICASSP.2015.7178964.

Zhiliang Peng, Jianwei Yu, Wenhui Wang, Yaoyao Chang, Yutao Sun, Li Dong, Yi Zhu, Weijiang Xu, Hangbo Bao, Zehua Wang, Shaohan Huang, Yan Xia, and Furu Wei. Vibevoice technical report. *arXiv preprint arXiv:2508.19205*, 2025. URL https://arxiv.org/abs/2508.19205.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine Mcleavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 28492–28518. PMLR, 2023. URL https://proceedings.mlr.press/v202/radford23a.html.

Simon Rouard, Manu Orsini, Axel Roebel, Neil Zeghidour, and Alexandre Défossez. Continuous audio language models. *arXiv preprint arXiv:2509.06926*, 2025. URL https://arxiv.org/abs/2509.06926.

Takaaki Saeki, Detai Xin, Wataru Nakata, Tomoki Koriyama, Shinnosuke Takamichi, and Hiroshi Saruwatari. UTMOS: UTokyo-SaruLab system for VoiceMOS challenge 2022. In *Interspeech 2022*, pp. 4521–4525, 2022. doi: 10.21437/Interspeech.2022-439. URL https://www.isca-archive.org/interspeech_2022/saeki22c_interspeech.html.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *The Tenth International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=TIdIXIpzhoI.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32211–32252. PMLR, 2023. URL https://proceedings.mlr.press/v202/song23a.html.

Yutao Sun, Hangbo Bao, Wenhui Wang, Zhiliang Peng, Li Dong, Shaohan Huang, Jianyong Wang, and Furu Wei. Multimodal latent language modeling with next-token diffusion. *arXiv preprint arXiv:2412.08635*, 2024. URL https://arxiv.org/abs/2412.08635.

Arnon Turetzky, Nimrod Shabtay, Slava Shechtman, Hagai Aronowitz, David Haws, Ron Hoory, and Avihu Dekel. Continuous speech synthesis using per-token latent diffusion. *arXiv preprint arXiv:2410.16048*, 2024. URL https://arxiv.org/abs/2410.16048.

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. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111*, 2023a. URL https://arxiv.org/abs/2301.02111.

Hongji Wang, Chengdong Liang, Shuai Wang, Zhengyang Chen, Binbin Zhang, Xu Xiang, Yanlei Deng, and Yanmin Qian. WeSpeaker: A research and production oriented speaker embedding learning toolkit. In *2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2023b. doi: 10.1109/ICASSP49357.2023.10096626. URL https://doi.org/10.1109/ICASSP49357.2023.10096626.

Hui Wang, Shujie Liu, Lingwei Meng, Jinyu Li, Yifan Yang, Shiwan Zhao, Haiyang Sun, Yanqing Liu, Haoqin Sun, Jiaming Zhou, Yan Lu, and Yong Qin. FELLE: Autoregressive speech synthesis with token-wise coarse-to-fine flow matching. In *Proceedings of the 33rd ACM International Conference on Multimedia*, 2025a. doi: 10.1145/3746027.3755494. URL https://arxiv.org/abs/2502.11128.

Wei Wang, Rong Cao, Yi Guo, Zhengyang Chen, Kuan Chen, and Yuanyuan Huo. Intmeanflow: Few-step speech generation with integral velocity distillation. *arXiv preprint arXiv:2510.07979*, 2025b. URL https://arxiv.org/abs/2510.07979.

Chun Yat Wu, Jiajun Deng, Guinan Li, Qiuqiang Kong, and Simon Lui. CLEAR: Continuous latent autoregressive modeling for high-quality and low-latency speech synthesis. *arXiv preprint arXiv:2508.19098*, 2025. URL https://arxiv.org/abs/2508.19098.

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. URL https://arxiv.org/abs/2509.02020.

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. *arXiv preprint arXiv:2506.21619*, 2025a. URL https://arxiv.org/abs/2506.21619.

Yixuan Zhou, Guoyang Zeng, Xin Liu, Xiang Li, Renjie Yu, Ziyang Wang, Runchuan Ye, Weiyue Sun, Jiancheng Gui, Kehan Li, Zhiyong Wu, and Zhiyuan Liu. VoxCPM: Tokenizer-free TTS for context-aware speech generation and true-to-life voice cloning. *arXiv preprint arXiv:2509.24650*, 2025b. URL https://arxiv.org/abs/2509.24650.

Yixuan Zhou, Guoyang Zeng, Xin Liu, Xiang Li, Renjie Yu, Jiancheng Gui, Jiaheng Wu, Ziyang Wang, Xudong Shen, Runchuan Ye, Zhisheng Zhang, Jiuyang Zhou, Bingsong Bai, Weiyue Sun, Mengyuan Deng, Qundong Shi, Zhiyong Wu, and Zhiyuan Liu. VoxCPM2 technical report. *arXiv preprint arXiv:2606.06928*, 2026. URL https://arxiv.org/abs/2606.06928.

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

A Training and Inference Details

A.1 Model Configuration

Tables 5 and 6 summarize the model configurations used in our experiments. The audio VAE accepts and reconstructs 24-kHz waveforms. It uses a causal convolutional architecture and a σ-VAE posterior with its standard deviation fixed to 0.15. CuteTTS and CuteTTS-distill share the same overall architecture, while the latter augments the diffusion head with guidance-strength and step-size embeddings. Both models group two latent frames per patch, resulting in an LM token rate of 6.25 tokens per second.

A.2 Training Configuration

Audio VAE. Training is conducted in FP32 for 1M steps with an effective batch size of 256. Training samples are randomly cropped into 2.5-s segments. To improve reconstruction from bandwidth-degraded speech, we downsample 5% of the training examples to 16 kHz and another 5% to 8 kHz, resample them back to 24 kHz, and retain the original 24-kHz signals as reconstruction targets. The loss weights in Equation 2 are $\lambda_{\text{mel}} = 15$, $\lambda_{\text{adv}} = 1$, $\lambda_{\text{feat}} = 2$, $\lambda_{\text{KL}} = 0.1$, and $\lambda_{\text{sem}} = 1$. We use AdamW with a learning rate of 2×10^{-4}, $(\beta_1, \beta_2) = (0.8, 0.99)$, and weight decay of 0.01, followed by exponential decay with $\gamma = 0.999996$.

CuteTTS. We train the TTS backbone end-to-end for 1M steps with a global batch size of up to 81,920 packed tokens. For flow-matching training, each target patch is repeated four times; each copy uses independently sampled Gaussian noise and $t = \text{sigmoid}(u)$, where $u \sim \mathcal{N}(0, 1)$. Conditional inputs are dropped with probability 0.1 to enable CFG at inference. We use AdamW with a peak learning rate of 5×10^{-4}, $(\beta_1, \beta_2) = (0.9, 0.95)$, and weight decay of 0.01. The learning rate follows a cosine schedule with 5K warmup steps.

CuteTTS-distill. We update only the diffusion head for 100K steps with a global batch size of up to 65,536 packed tokens. Each target patch is repeated twice for distillation. Teacher targets are computed on an eight-step base grid. At 0, 20K, and 50K training steps, the sampling probabilities over four-, two-, and one-step targets are $(1, 0, 0)$, $(0.5, 0.5, 0)$, and $(0.2, 0.3, 0.5)$, respectively; the interval start is sampled uniformly from the selected discrete grid. We use AdamW with a peak learning rate of 1×10^{-5}, $(\beta_1, \beta_2) = (0.9, 0.95)$, weight decay of 0.01, and a cosine schedule with 1K warmup steps. The resulting checkpoint supports all three inference budgets.

Table 5: Audio VAE configuration.

Hyperparameter	Value
Frame rate (Hz)	12.5
Latent dimension	64
Encoder initial channels	128
Encoder strides	3/5/8/16
Decoder initial channels	1,536
Decoder strides	16/8/5/3
Parameters (M)	127.9

A.3 Inference Configuration

Voice-cloning CFG. The base model uses LM-level CFG with guidance weight $w = 2$. Table 7 summarizes the two branches. Both branches retain the generated patch history. The unconditional branch removes the target text and reference audio from the LM input and zeros the diffusion-head speaker condition. CFG therefore jointly strengthens the LM-side text and reference conditions and the diffusion-head speaker condition.

Sampling. For each generated patch, CuteTTS uses 10 diffusion-head NFEs on each CFG branch, with Euler integration over a Sway time grid with coefficient -0.8. CuteTTS-distill encodes $w = 2$

Table 6: Model configurations. Parameter counts are rounded.

Component	Hyperparameter	**CuteTTS**	**CuteTTS-distill**
Patch encoder	Layers	2	2
	Hidden dimension	1,024	1,024
	Attention/KV heads	16/2	16/2
	FFN dimension	4,096	4,096
	Parameters (M)	31.0	31.0
Causal backbone	Layers	7	7
	Hidden dimension	1,024	1,024
	Attention/KV heads	16/8	16/8
	FFN dimension	3,072	3,072
	Parameters (M)	126.9	126.9
Diffusion head	Layers	4	4
	Hidden dimension	1,024	1,024
	Attention/KV heads	16/2	16/2
	FFN dimension	4,096	4,096
	Guidance-strength embedding	–	✓
	Step-size embedding	–	✓
	Parameters (M)	70.5	73.6
Overall	Parameters (M)	228.6	231.8

Table 7: LM-CFG branches for zero-shot voice cloning with CuteTTS.

Branch	LM input	Diffusion-head speaker
Conditional	⟨reference audio⟩ | ⟨text⟩	$\boldsymbol{s}$
Unconditional	Unconditional LM prefix	$\mathbf{0}$

directly as a diffusion-head condition and uses four NFEs on a single uniform Euler grid, without an unconditional branch. The same distilled checkpoint supports one-, two-, and four-step inference; unless otherwise noted, the main experiments use four steps.

A.4 Ablation Configuration

VAE. The VAE ablations accept and reconstruct 24-kHz waveforms and use the same causal convolutional architecture, 64-dimensional σ-VAE posterior, and fixed posterior standard deviation of 0.15 as the main audio VAE. As summarized in Table 8, the four frame-rate variants differ only in the final encoder downsampling factor and the corresponding decoder upsampling factor. All other model settings are shared.

The four variants are trained for 500K steps in FP32 on randomly cropped 2.5-s segments with an effective batch size of 64. We optimize both the generator and discriminator using AdamW with a learning rate of 1×10^{-4}, $(\beta_1, \beta_2) = (0.8, 0.99)$, and weight decay of 0.01, followed by exponential decay with $\gamma = 0.999996$. The weights for mel reconstruction, adversarial feature matching, adversarial generation, KL regularization, and semantic alignment are 15, 2, 1, 0.1, and 1, respectively.

TTS. Except for the module-wise capacity study in Appendix B.3, all TTS ablations use the model configuration in Table 6. The layer counts and hidden dimensions are held fixed across the Rate–Patch, semantic-alignment, and speaker-conditioning experiments, while the patch-encoder input and diffusion-head output projections follow the patch size p.

These models are trained end-to-end for 500K steps with a global batch size of up to 40,960 packed tokens. We use the same conditional flow-matching setup as for CuteTTS, including four independently noised copies of each target patch. AdamW is used with a peak learning rate of 5×10^{-4}, $(\beta_1, \beta_2) = (0.9, 0.95)$, and weight decay of 0.01. The learning rate follows a cosine

Table 8: Model configurations for the VAE frame-rate ablation.

Hyperparameter	6.25 Hz	12.5 Hz	25 Hz	50 Hz
Latent dimension	64	64	64	64
Encoder initial channels	128	128	128	128
Encoder strides	3/5/8/32	3/5/8/16	3/5/8/8	3/5/8/4
Decoder initial channels	1,536	1,536	1,536	1,536
Decoder strides	32/8/5/3	16/8/5/3	8/8/5/3	4/8/5/3
Parameters (M)	232.7	127.9	75.5	49.2

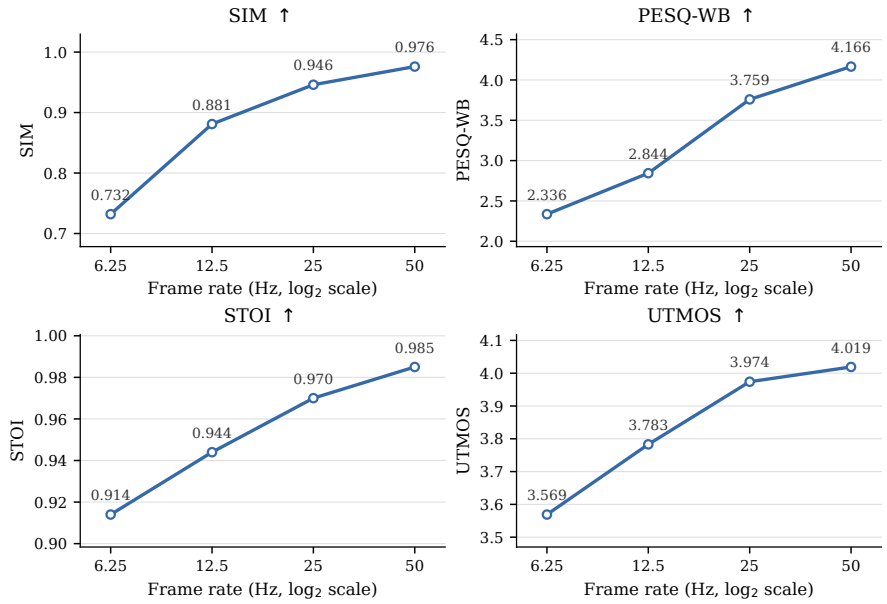


Figure 5: VAE reconstruction quality at different frame rates on LibriSpeech test-clean. All models use the same training data and training configuration.

schedule with 5K warmup steps. To enable CFG at inference, conditional inputs are dropped with probability 0.1 during training.

B Temporal Granularity and Capacity Analysis

The VAE frame rate r and patch size p jointly determine the temporal resolution of the speech representation and the granularity of autoregressive prediction. Given a VAE frame rate of r frames per second and p frames per patch, the LM token rate q, defined as the number of autoregressive steps taken by the causal backbone per second, is

$$q = \frac{r}{p}. \tag{12}$$

A higher r provides finer temporal detail but increases the latent sequence length, whereas a larger p lowers the LM token rate but requires the diffusion head to generate a larger continuous patch at each step. Different combinations of r and p can yield the same LM token rate while differing in representation resolution and patch-generation difficulty. We therefore first examine two controlled settings: varying r with p fixed, and varying r and p while holding q constant. We then complement these comparisons with module-wise depth scaling to examine how different temporal granularities interact with the capacities of the causal backbone and diffusion head.

Table 9: TTS performance at a fixed patch size of $p = 1$ on LibriSpeech test-clean.

Model	WER (%)↓	SIM↑	UTMOS↑
$\mathrm{TTS}_{6.25,1}$	**2.53**	62.5	3.733
$\mathrm{TTS}_{12.5,1}$	4.67	**71.0**	**3.883**
$\mathrm{TTS}_{25,1}$	10.59	**71.0**	3.833
$\mathrm{TTS}_{50,1}$	70.33	67.9	3.346

Table 10: TTS performance at a fixed LM token rate of $q = 6.25$ tokens per second on LibriSpeech test-clean.

Model	WER (%)↓	SIM↑	UTMOS↑
$\mathrm{TTS}_{6.25,1}$	**2.53**	62.5	3.733
$\mathrm{TTS}_{12.5,2}$	2.70	71.3	**3.878**
$\mathrm{TTS}_{25,4}$	3.92	72.5	3.749
$\mathrm{TTS}_{50,8}$	7.04	**74.1**	3.649

B.1 Representation Rate and Reconstruction Quality

To study the effect of the VAE frame rate r, we train four VAEs with $r \in \{6.25, 12.5, 25, 50\}$ Hz using the same training data and training configuration. The models differ only in the downsampling factor of the encoder's final stage and the corresponding upsampling factor of the decoder's final stage. Their reconstruction performance on LibriSpeech test-clean is visualized in Figure 5. Reconstruction quality improves consistently as r increases across all four metrics. The improvement is steepest from 6.25 Hz to 12.5 Hz and gradually diminishes at higher frame rates.

B.2 Joint Effects of VAE Frame Rate and Patch Size

We denote a TTS model with VAE frame rate r and patch size p as $\mathrm{TTS}_{r,p}$, where r is measured in Hz and p is the number of VAE frames in each patch. Using a training configuration shared across all variants, we train the configurations reported below and evaluate them on LibriSpeech test-clean.

Fixed patch size. We first fix $p = 1$, for which the LM token rate grows directly with the VAE frame rate. Table 9 shows that increasing r initially improves SIM and UTMOS, consistent with the stronger reconstruction capability of the higher-rate VAEs. However, WER rises rapidly because the causal backbone must make proportionally more autoregressive decisions. At 50 Hz, WER increases sharply and both SIM and UTMOS decline relative to the intermediate-rate models, suggesting that the resulting temporal granularity exceeds the modeling capacity of the fixed backbone.

Fixed LM token rate. To separate autoregressive sequence length from patch-generation difficulty, we next fix the LM token rate at $q = r/p = 6.25$ tokens per second. The four configurations in Table 10 require the same number of autoregressive decisions but use progressively higher VAE frame rates and larger patches. $\mathrm{TTS}_{12.5,2}$ improves SIM and UTMOS substantially over $\mathrm{TTS}_{6.25,1}$ with only a small change in WER, showing that a moderate patch size can exploit the stronger reconstruction capability of a higher-rate VAE without materially degrading intelligibility. With still larger patches, SIM continues to improve, but WER increases and UTMOS declines. This divergence shows that LM token rate alone does not characterize synthesis difficulty: larger patches reduce the autoregressive sequence length but alter the relative demands on inter-patch autoregressive modeling and within-patch continuous generation. We next examine this interaction through module-wise capacity scaling.

B.3 Module-wise Capacity Sensitivity

The preceding experiments show that temporal granularity affects both autoregressive sequence modeling and within-patch generation. To further understand how these two modeling demands interact with model capacity, we consider two configurations using the same 50-Hz speech representation but

Table 11: Model configurations for the module-wise capacity study. Parameter counts include the TTS model only.

Model	Causal backbone layers	Diff. head layers	Params. (M)
$\mathrm{TTS}_{50,1}$	7	4	228.6
Causal backbone $\times 2$	14	4	338.8
Diffusion head $\times 2$	7	8	294.7
$\mathrm{TTS}_{50,8}$	7	4	228.6
Causal backbone $\times 2$	14	4	338.8
Diffusion head $\times 2$	7	8	294.7

Table 12: Module-wise depth scaling under two temporal granularities with a fixed 50-Hz speech representation. "Causal backbone $\times 2$" and "Diffusion head $\times 2$" independently double the depth of the corresponding module.

Model	WER (%)↓	SIM↑	UTMOS↑
$\mathrm{TTS}_{50,1}$	70.33	67.9	3.346
Causal backbone $\times 2$	**60.03**	72.1	**3.434**
Diffusion head $\times 2$	90.84	**72.4**	3.419
$\mathrm{TTS}_{50,8}$	7.04	74.1	3.649
Causal backbone $\times 2$	**4.91**	74.4	3.700
Diffusion head $\times 2$	5.85	**76.8**	**3.924**

substantially different patch sizes, $\mathrm{TTS}_{50,1}$ and $\mathrm{TTS}_{50,8}$. They respectively represent a fine-grained regime with a small patch and a long autoregressive sequence, and a coarse-grained regime with a large patch and a much shorter autoregressive sequence. Starting from the base architecture, we independently double the depth of either the causal backbone or the diffusion head while keeping the other component unchanged. Table 11 summarizes the resulting module depths and parameter counts. All variants use the same training data, optimization schedule, and inference configuration as the standard TTS ablations described in Appendix A.4. Since doubling the depth of the two modules results in different increases in parameter count, we interpret this experiment as a module-wise sensitivity analysis rather than a parameter-matched capacity comparison.

Table 12 shows that, for the fine-grained $\mathrm{TTS}_{50,1}$ configuration, the causal backbone operates at 50 autoregressive steps per second. Doubling its depth reduces WER from 70.33% to 60.03%, while also improving SIM and UTMOS. In contrast, enlarging the diffusion head improves speaker similarity but does not alleviate the severe intelligibility degradation, with WER further increasing to 90.84%. This suggests that, when autoregressive prediction is extremely fine-grained, the long sequence places substantial modeling pressure on the causal backbone, and additional local-generation capacity alone cannot compensate for this difficulty.

A different pattern emerges for $\mathrm{TTS}_{50,8}$, where patching reduces the LM token rate from 50 to 6.25 tokens per second. Increasing causal-backbone depth primarily improves intelligibility, reducing WER from 7.04% to 4.91%, while producing only marginal gains in SIM and UTMOS. In contrast, doubling the diffusion-head depth produces substantially larger improvements in speaker fidelity and predicted perceptual quality: SIM increases from 74.1 to 76.8 and UTMOS from 3.649 to 3.924, while WER also improves to 5.85%. This indicates that, once the autoregressive sequence is substantially shortened and each autoregressive step is responsible for a larger continuous patch, additional diffusion-head capacity becomes increasingly useful for local acoustic generation.

Taken together, the two configurations exhibit complementary module-wise capacity sensitivities. Increasing causal-backbone capacity primarily benefits intelligibility, consistent with its role in

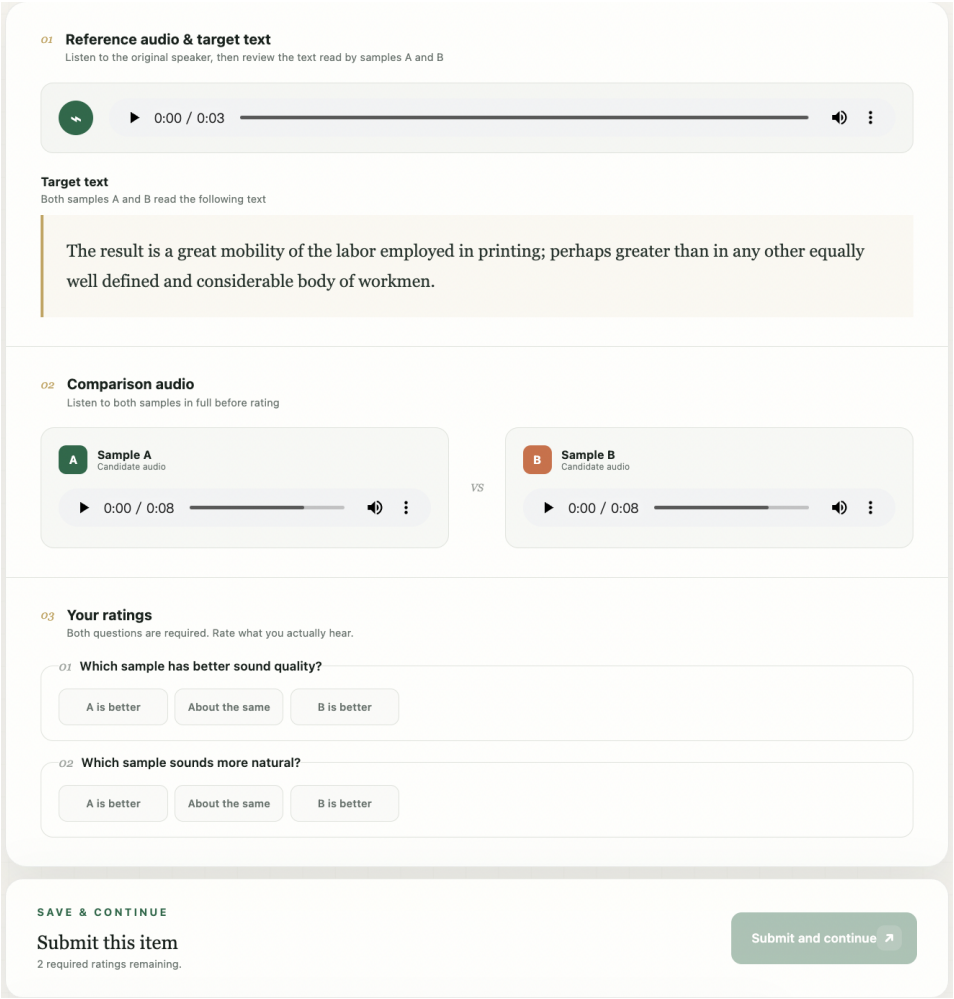


Figure 6: Annotation interface for the subjective evaluation.

modeling inter-patch dependencies, whereas increasing diffusion-head capacity more strongly benefits speaker fidelity and perceptual quality, consistent with its role in within-patch continuous generation. Temporal granularity further modulates the relative importance of these capacities: causal-backbone capacity is particularly important for fine-grained autoregressive prediction, while diffusion-head capacity becomes more valuable under large-patch generation. Temporal granularity therefore controls more than the autoregressive token rate: it also changes how modeling capacity is utilized across inter-patch autoregression and within-patch continuous generation.

C Subjective Evaluation Details

We conduct the subjective evaluation on the 50-example LibriSpeech test-clean subset. Each of the two CuteTTS variants is compared with MOSS-TTS-Nano, F5-TTS, ZipVoice, and Pocket TTS on all 50 examples, yielding 400 comparison items in total. The items are randomly shuffled and distributed among seven annotators. For each pair of generated samples, annotators select the better sample independently for sound quality and naturalness, with a tie permitted for either dimension. System identities are hidden throughout the evaluation. Figure 6 shows the annotation interface.

D EFFICIENCY BENCHMARK

All systems are evaluated on the fixed 50-request LibriSpeech test-clean subset using a single NVIDIA RTX 4090. We keep one model resident at a time and process requests strictly sequentially with batch size one. Measurements follow a warm-service, warm-file-cache protocol: each model completes one full warm-up pass before measurement, while model loading, compilation, and warm-up are excluded from the timed interval.

Each voice-cloning request begins with raw target text and a raw reference-audio path. Timing starts before text processing and reference file access and includes, where applicable, audio loading and resampling, reference and speaker encoding, LM prefill, latent generation, causal VAE decoding, and device-to-host transfer. No request-level tokenizer output, decoded reference audio, reference representation, speaker embedding, or prefix state is reused across requests; only the operating-system file cache is retained. Table 2 reports the arithmetic mean, P50, and P95 first-audio latency, together with the mean per-utterance RTF, over the fixed 50 requests.

For CuteTTS, the inference configurations follow Appendix A. The base and distilled models use the same request order and per-item seeds. Their fixed-shape diffusion heads and Euler sampling loops are compiled, while CUDA graphs are disabled. Both models successfully complete all 50 requests without truncation.

For baseline inference configurations, MOSS-TTS and Qwen3-TTS use the official vLLM-Omni inference engine, while ZipVoice uses the official PyTriton/TensorRT engine. The MOSS-TTS Delay model accepts a streaming request but returns a single final PCM block; we therefore report only its RTF. MOSS-TTS-Nano is evaluated in raw-reference voice-cloning mode. Since VibeVoice-Realtime does not support arbitrary-reference voice cloning, it is evaluated using its default voice.