CaDiS: Causal Diffusion Transformer for Streaming Speech Generation

Anonymous authors
Paper under double-blind review

Abstract

Diffusion and flow-matching models have advanced high-fidelity speech generation. However, their bidirectional attention requires generating a full utterance before playback, and iterative sampling adds further delay. Most streaming systems therefore rely on autoregressive generators and use diffusion only for acoustic decoding. In this paper, we ask instead whether a pretrained offline diffusion model can generate speech chunk by chunk in real time. We present **CaDiS**, an offline-to-streaming framework built on a causal diffusion transformer that decouples the architectural and acceleration challenges into a three-stage recipe: autoregressive diffusion adaptation finetunes the bidirectional model with block-causal attention to generate each chunk from past speech alone; causal consistency distillation reduces sampling to four steps; and distribution matching refinement restores teacher-level quality. We adapt CaDiS to two streaming tasks, text-to-speech (TTS) and voice conversion (VC), with dedicated task-specific efforts: a normalized positional encoding for asynchronously aligned text in TTS, and content-vector distillation plus reference-anchored positional re-encoding to address the voice consistency and artifact challenges in VC. We evaluate CaDiS on objective and subjective metrics across natural human and expressive anime voices. CaDiS preserves offline-teacher quality while streaming: its TTS model attains a 286 ms first-packet latency and 0.31 real-time factor on an A100 GPU, while its VC model achieves the best intelligibility and perceptual quality among streaming baselines with no look-ahead. Moreover, CaDiS attains among the best overall streaming efficiency profiles, demonstrating the potential of diffusion speech generators in streaming scenarios.

1 Introduction

Recent advances in generative modeling have driven remarkable progress in speech generation, producing results of high naturalness, intelligibility, and faithfulness (Du et al., 2024; Chen et al., 2025; Liu, 2024). Building on this progress, streaming speech generation has shown growing promise for interactive applications such as audio previewing, conversational voice agents, and instant voice design. Unlike offline generation, however, it imposes stricter requirements on latency, playback pacing, and the ability to sustain generation in open-ended conversation.

Diffusion models have emerged as leading high-fidelity speech generators. Different from autoregressive (AR) generators built on discrete acoustic tokens, they model continuous acoustic representations, thereby avoiding the fidelity loss introduced by the vector quantization that code-based AR systems adopt for compression (Meng et al., 2025). Together with the iterative refinement, diffusion models yield strong naturalness, speaker similarity, and expressiveness (Le et al., 2023; Ju et al., 2024; Eskimez et al., 2024; Chen et al., 2025). However, such strengths come at the cost of streamability: the bidirectional architectures typically require the full acoustic context, making incremental generation challenging, and the iterative sampling adds further delay. As a result, current streaming systems remain dominated by AR generators, where diffusion is relegated to local acoustic refinement at the tail of the pipeline (Du et al., 2024; 2025; Guo et al., 2026; Choi & Lee, 2025; Chen et al., 2024b; Anastassiou et al., 2024). This leaves the quality advantages of diffusion underused and raises a central question: *can a pretrained diffusion model itself generate speech chunk by chunk at low latency, without sacrificing its offline quality?*

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.

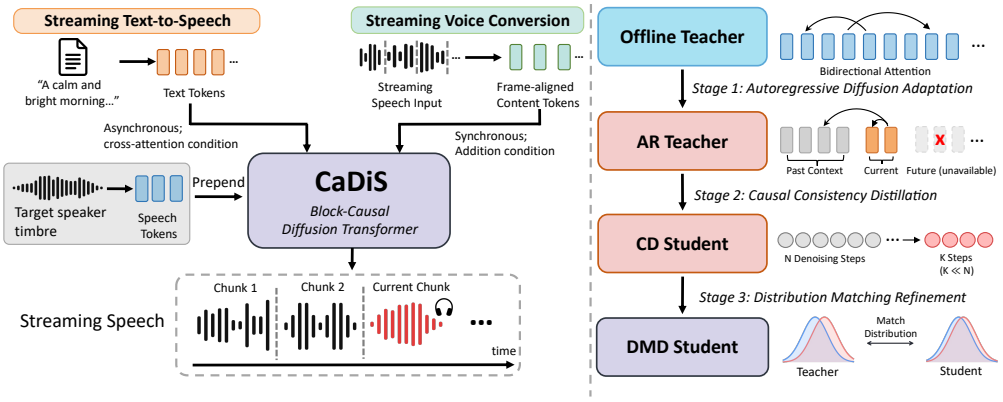


Figure 1: The proposed CaDiS architecture and the three-stage recipe. Left: how CaDiS serves streaming TTS and VC tasks. Right: the three-stage recipe to adapt the offline teacher through autoregressive diffusion adaptation (AR Teacher), causal consistency distillation (CD Student) and distribution matching refinement (DMD Student as the final streaming model).

Adapting an offline diffusion model for streaming poses three challenges: **causality**, **sampling efficiency**, and **long-form stability**. Causality requires generating each speech chunk without future acoustic context. Sampling efficiency requires reducing the number of denoising steps. Long-form stability requires the model to sustain quality over continuous, potentially unbounded streams rather than fixed-length utterances. These challenges take task-specific forms, because TTS and VC differ in how their conditioning aligns with the output. In TTS, text and speech are asynchronously aligned, so the model must maintain global text-speech alignment across chunks. In VC, the source content is frame-synchronous with the output and arrives continuously, so the model must preserve acoustic continuity and speaker identity throughout an open-ended stream using only bounded past context. A streaming framework must therefore address all three challenges while respecting the conditioning structure of each task.

We present **CaDiS**, which adapts a pretrained diffusion transformer (DiT) (Peebles & Xie, 2023) speech generator for low-latency streaming while keeping DiT as its generative core (Figure 1). CaDiS addresses the streaming-adaptation challenges above through a three-stage recipe. First, autoregressive diffusion adaptation finetunes the pretrained DiT with block-causal attention so that each chunk is generated from past speech alone. Next, causal consistency distillation reduces the denoising process to four steps. Finally, distribution matching refinement restores the distilled streaming model to teacher-level generation quality. The recipe builds on Causal Forcing++ (Zhao et al., 2026), originally developed for video generation, which we adapt to speech while addressing noticeable and distinct temporal conditioning structures of speech tasks. For TTS, we introduce a normalized positional encoding to accommodate the asynchronous text-speech alignment. For VC, we distill a streamable content-vector encoder (Qian et al., 2022) that produces frame-synchronous content features aligned with the acoustic latents. And we further adapt reference-anchored positional re-encoding (RAPR) (Sun et al., 2026) by pairing a long speaker reference with a short rolling history to sustain speaker identity from bounded context over streams. Audio samples can be found at https://cadis.pages.dev.

Our contributions are three-fold:

- To our knowledge, CaDiS is among the first streamable DiT speech generators operating on continuous latents. The three-stage recipe adapts a pretrained offline model for both streaming TTS and VC tasks.
- We develop dedicated task-specific designs on CaDiS for streaming TTS and VC tasks: a globally normalized positional encoding for TTS, and reference-anchored positional re-encoding (RAPR) together with content-vector distillation for VC to mitigate the voice consistency and artifact issues.
- Extensive objective and subjective evaluations demonstrate that CaDiS produces 48 kHz audio with competitive overall quality on TTS and the best overall quality on VC among streaming baselines, while achieving a competitive streaming efficiency profile in terms of first-packet latency and real-time factor.

2 Related Work

Diffusion and flow-matching speech generation. Continuous-latent diffusion and flow models have become strong offline speech generators (Le et al., 2023; Ju et al., 2024; Eskimez et al., 2024; Chen et al., 2025). They assume bidirectional access to the full utterance and iterative sampling. How to retain their performance under causal, few-step inference therefore remains underexplored.

Streaming text-to-speech. Most streaming TTS systems are autoregressive or hybrid, using a language model to predict speech tokens followed by a local acoustic decoder (Chen et al., 2024b; Anastassiou et al., 2024; Du et al., 2024; 2025). Recent work makes this decoding stage streamable via chunk or block-wise flow-matching (Du et al., 2024; Guo et al., 2026; Choi & Lee, 2025), while Dragon-FM combines autoregressive generation across codec-token chunks with flow matching within each chunk (Liu et al., 2025). CaDiS shares this block-autoregressive perspective, but starts from a bidirectional continuous-latent diffusion generator and directly causalizes it into a few-step streaming model, generating speech from full-sentence text rather than jointly consuming incremental text and emitting audio (Dang et al., 2024; Sheng et al., 2026; Torgashov et al., 2026).

Streaming voice conversion. Existing streaming VC systems typically rely on causal content encoders, streaming-aware training, or language-model conditioning outside the diffusion family (Yang et al., 2024; Ning et al., 2023; 2024; Wang et al., 2024). Diffusion and flow models have only recently entered this setting: Seed-VC offers a streaming mode but does not turn an offline generator into a block-causal, few-step primary streamer, while X-VC and MeanVC-2 cut latency using short future buffers (Liu, 2024; Zheng et al., 2026; Ma et al., 2026). Unlike TTS, VC has no sentence boundaries, so frame-level content alignment and stable speaker identity must hold over long rollouts. Whether zero-look-ahead VC and streaming TTS can share a single causal generator design thus remains open.

Causal and few-step diffusion. Diffusion Forcing and rolling diffusion introduce temporally structured noise for sequential generation (Chen et al., 2024a; Ruhe et al., 2024). CausVid and Self-Forcing distill bidirectional video models into block-causal generators (Yin et al., 2025; Huang et al., 2025). Causal Forcing++ most directly motivates our three-stage recipe (Zhao et al., 2026). In speech, ARDiT and DiTAR instead train autoregressive diffusion generators directly (Liu et al., 2024; Jia et al., 2025), while consistency and distribution-matching methods target few-step synthesis rather than streaming (Ye et al., 2023; 2024; Li et al., 2025). We ask whether causal distillation can adapt a strong offline speech DiT to both tasks. CaDiS addresses these concerns via task-specific designs.

3 Method

Figure 1 illustrates the architecture of CaDiS and the three-stage recipe used to train it. CaDiS adapts a pretrained offline DiT speech generator into a streaming one. The backbone is shared across TTS and VC tasks, but we train a separate task-specific instance for each, distilled from its corresponding offline teacher. We train CaDiS through three stages: autoregressive diffusion adaptation, causal consistency distillation, and distribution matching refinement. This section is organized as follows: the streaming generation problem formulation in Section 3.1; the architecture of the offline speech generator in Section 3.2; the details of the three-stage causalization procedure in Section 3.3; and the task-specific designs for TTS and VC in Sections 3.4–3.6.

3.1 Problem formulation

Let $\mathbf{z} = (\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(K)})$ denote a clean audio latent sequence partitioned into K chunks of L frames, where $\mathbf{z}^{(k)}$ is the k-th chunk and $\mathbf{z}^{(<k)}$ denotes all preceding chunks. Let $\mathbf{r}$ be a target-speaker reference and $\mathbf{u}$ the task-dependent content input. While the offline teacher p_{off} models the full latent sequence bidirectionally, a streaming model generates it chunk by chunk as:

$$q_\theta(\mathbf{z} \mid \mathbf{r}, \mathbf{u}) = \prod_{k=1}^{K} q_\theta\left(\mathbf{z}^{(k)} \mid \mathbf{r}, \mathcal{H}_k, \mathbf{u}_k\right), \qquad \mathcal{H}_k = \mathcal{C}_k(\mathbf{z}^{(<k)}). \tag{1}$$

Here, $\mathcal{C}_k$ selects the previously generated audio context, and $\mathbf{u}_k$ denotes the content available to chunk k. For TTS, $\mathbf{u}_k$ is the text of the current sentence, which is available in full before generation begins. For VC, it is the source-content representation aligned with chunk k. Our goal is to learn these causal chunk conditionals such that their streaming rollout approximates the offline distribution.

3.2 OFFLINE DIFFUSION SPEECH GENERATOR

Our pretrained offline generator is a stack of DiT blocks (Peebles & Xie, 2023) operating on 40 Hz continuous latents encoded from 48 kHz mono audio by a frozen DAC-VAE trained with target-KL regularization (Seetharaman & Kumar, 2026). Both TTS and VC tasks share two conditioning signals: a *speaker reference* that specifies the target timbre, and a *content reference* that specifies what is spoken. For both tasks, the speaker reference is a speech clip that is encoded into audio latents and prepended to the output latent sequence (excluded from the training loss). The content reference is task-specific. For TTS, the content reference is text transcript, which is encoded by an mT5-large text encoder (Xue et al., 2021) into text tokens for cross-attention condition of the DiT blocks. For VC, the content reference is another speech utterance, which is encoded by a bidirectional 50 Hz ContentVec encoder (Qian et al., 2022), resampled to 40 Hz to match the output latent frame rate, and added frame-wise to the output latent sequence. Note that the content reference differs fundamentally between the two tasks: for TTS it is asynchronous text specifying only the linguistic content, whereas for VC it is a frame-aligned representation and the VC model must follow the content frame-by-frame.

The model is trained with a conditional flow-matching objective (Lipman et al., 2023) that regresses the velocity field of a variance-preserving and shifted-cosine probability path. Training uses classifier-free guidance (Ho & Salimans, 2022), and the generated latents are decoded to audio by DAC-VAE.

3.3 FROM BIDIRECTIONAL TO CAUSAL DIFFUSION

The offline model is trained with bidirectional attention over the full acoustic context. Simply applying a causal mask alters the conditional distribution to Equation 1, and rolling the masked model out over many chunks compounds this mismatch into audible drift. To address this, we first adapt the block-causal architecture and then accelerate its sampling by reducing the number of function evaluations (NFEs), with each stage below serving a distinct role.

Stage 1: Autoregressive Diffusion Adaptation. Stage 1 addresses the architectural gap to meet the streaming requirement. We replace the bidirectional mask with a block-causal mask and finetune the DiT to predict each chunk from *clean preceding chunks* (i.e., teacher-forcing). Each teacher-forcing example is arranged as $[\,\mathbf{r} \mid \mathbf{z} \mid \mathbf{z}_t\,]$, where $\mathbf{r}$ is the speaker reference, and $\mathbf{z}$ and $\mathbf{z}_t$ are clean and noisy speech sequences of the *same* target utterance. For chunk k, we use the noise schedule (α_t, σ_t) inherited from the offline teacher, to construct the noisy input and velocity target with noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and optimize the objective:

$$\mathcal{L}_{\mathrm{AR}}(\phi) = \mathbb{E}_{k,t,\boldsymbol{\epsilon}}\left[\left\|\mathbf{v}_\phi(\mathbf{z}_t^{(k)}, t \mid \mathbf{c}_k) - \mathbf{v}_t^{*(k)}\right\|_{\mathbf{M}_k}^2\right] \;\middle|\; \begin{array}{r} \mathbf{z}_t^{(k)} = \alpha_t \mathbf{z}^{(k)} + \sigma_t \boldsymbol{\epsilon}, \\ \mathbf{v}_t^{*(k)} = \alpha_t \boldsymbol{\epsilon} - \sigma_t \mathbf{z}^{(k)}, \\ \mathbf{c}_k \triangleq (\mathbf{r}, \mathcal{H}_k, \mathbf{u}_k). \end{array} \tag{2}$$

Where t denotes diffusion time throughout, ϕ denotes the parameters of this block-causal multi-step model, $\mathbf{M}_k$ and $\mathbf{M}$ are binary validity masks for chunk k and the full utterance, respectively. We use $\|\cdot\|_{\mathbf{M}_k}$ and $\mathrm{mean}_{\mathbf{M}}$ to denote averages over valid and non-padded elements. $\mathbf{c}_k$ collects all conditioning visible to chunk k. We use a custom block-causal attention mask that preserves the clean preceding chunks required for teacher forcing while preventing future-context leakage. By the masking design, all chunks can be trained in parallel while ensuring that each prediction uses only the preceding context. The resulting model $\mathbf{v}_\phi$ learns teacher-forced causal chunk conditionals and has the same sampling cost as the offline teacher during inference.

Stage 2: Causal Consistency Distillation. Stage 2 closes the sampling-cost gap. We adapt Causal Forcing++'s online causal consistency objective (Song et al., 2023; Zhao et al., 2026) to distill the frozen causal teacher $\mathbf{v}_\phi$ into a four-step causal student $\mathbf{v}_\theta$. Initialized from the Stage 1 teacher, we define the corresponding clean-latent prediction of the student as $F_\theta(\mathbf{z}_t, t \mid \mathbf{c}_k) = \alpha_t \mathbf{z}_t - \sigma_t \mathbf{v}_\theta(\mathbf{z}_t, t \mid \mathbf{c}_k)$. For adjacent noise levels $t_i > t_{i-1}$, the teacher first denoises the sample by one deterministic ODE step $\Psi_\phi^{t_i \to t_{i-1}}$. We then match the student's clean prediction at t_i to that of its exponential-moving-average target $\bar{\theta}$ at t_{i-1}:

$$\mathcal{L}_{\mathrm{CCD}}(\theta) = \mathbb{E}\left[\left\|F_\theta(\mathbf{z}_{t_i}^{(k)}, t_i \mid \mathbf{c}_k) - \mathrm{sg}\left[F_{\bar{\theta}}(\widetilde{\mathbf{z}}_{t_{i-1}}^{(k)}, t_{i-1} \mid \mathbf{c}_k)\right]\right\|_{\mathbf{M}_k}^2\right] \;\middle|\; \widetilde{\mathbf{z}}_{t_{i-1}}^{(k)} = \Psi_\phi^{t_i \to t_{i-1}}\left(\mathbf{z}_{t_i}^{(k)} \mid \mathbf{c}_k\right). \tag{3}$$

Where $\bar{\theta}$ is an EMA copy of the student parameters θ, and sg denotes stop-gradient. Both branches share the same teacher-forced causal context, while only the current chunk is noisy. The resulting four-step student initializes Stage 3.

Stage 3: Distribution Matching Refinement. Stage 3 recovers the quality lost during few-step distillation. We perform DMD (Yin et al., 2024b;a) on student-forced rollouts. The causal student thus learns under its own generated context while matching the distribution of the offline teacher. Let G_θ denote the block-causal model parameterized by the Stage 2 student $\mathbf{v}_\theta$, and let $\widetilde{\mathbf{z}}_0 = G_\theta(\boldsymbol{\epsilon} \mid \mathbf{r}, \mathbf{u})$ be an utterance generated chunk by chunk. We re-noise the student rollout to $\widetilde{\mathbf{z}}_t$. Given the same noisy sample, the frozen offline teacher provides the real estimate $\widehat{\mathbf{z}}_{0,\mathrm{r}}$, while a fake-score model initialized from the offline teacher and updated on student samples provides the fake estimate $\widehat{\mathbf{z}}_{0,\mathrm{f}}$. We optimize the following loss:

$$\mathbf{g}_{\mathrm{DMD}} = \frac{\widehat{\mathbf{z}}_{0,\mathrm{f}}(\widetilde{\mathbf{z}}_t,t) - \widehat{\mathbf{z}}_{0,\mathrm{r}}(\widetilde{\mathbf{z}}_t,t)}{\mathrm{mean}_{\mathbf{M}}\left|\widetilde{\mathbf{z}}_0 - \widehat{\mathbf{z}}_{0,\mathrm{r}}\right|}, \qquad \mathcal{L}_{\mathrm{DMD}}(\theta) = \tfrac{1}{2}\mathbb{E}\left[\left\|\widetilde{\mathbf{z}}_0 - \mathrm{sg}[\widetilde{\mathbf{z}}_0 - \mathbf{g}_{\mathrm{DMD}}]\right\|_{\mathbf{M}}^2\right]. \tag{4}$$

After each chunk is generated, a detached copy is added to the causal cache, so subsequent chunks condition on student-generated rather than ground-truth history, without back-propagating through previously generated chunks. This matches the history seen during streaming inference while preserving four-step generation.

3.4 Streaming TTS

Text Condition and Global Alignment. For streaming TTS, the content condition is the raw text of *the current sentence*, injected into each acoustic chunk through cross-attention. We generate and stream the speech incrementally. Because the offline TTS model learns a global monotonic relationship between textual and acoustic progress, resetting the positional information during the generation process disrupts this alignment prior learned during pretraining. We therefore retain the normalized absolute positional encoding of our offline teacher when adapting to streaming. An external duration predictor estimates the total utterance length once from the sentence text, and each acoustic frame is then assigned its fractional position within this predicted length. This preserves sentence-level progress across streaming chunks and allows the block-causal model to maintain text-speech alignment without access to future acoustic context.

3.5 Streaming voice conversion

Temporally Synchronous Content Condition. To enable frame-synchronous conditioning, the content condition and the output latents must share the same frame rate. In our offline VC system, we interpolate the 50 Hz bidirectional ContentVec representation to 40 Hz to match the output latents. When adapting to a streaming model, we apply two optimizations: (1) we initialize a native 40 Hz ContentVec encoder for content condition, by relocating the convolutional strides of the pretrained 50 Hz encoder so that the same weights produce 40 Hz features; and (2) we make ContentVec block-causal by removing the look-ahead from both the convolutional and transformer layers. The resulting causal ContentVec encoder is trained to match the 50 Hz bidirectional ContentVec teacher (interpolated at 40 Hz output). As a result, the streaming VC model natively produces content and latent streams at the same rate, without a separate alignment step.

Long-form Streaming with RAPR. Unlike TTS, streaming VC has no sentence boundary to reset generation. The model must therefore preserve local prosodic continuity and target-speaker identity over an open-ended stream while keeping positional indices bounded. We adapt reference-anchored positional re-encoding (RAPR) (Sun et al., 2026) with a long speaker reference and a short rolling history. Let $\mathbf{r}$ be the reference, w the history-window length, and n_k the number of frames generated before chunk k. At chunk k, we keep the most recent $\min(n_k, w)$ frames as $\mathcal{R}_k^w$ and form the attention cache $\mathcal{K}_k^w = [\mathbf{r} \mid \mathcal{R}_k^w]$. Keys are cached before positional rotation, so positions can be reassigned as the window slides. The position map π_k preserves the reference anchors and re-indexes the retained history into the slots right after the reference. Once the stream exceeds w frames, the oldest frame is dropped, bounding the cache size and positional span at $|\mathbf{r}| + w$. Section 4.6 studies the effect of w length on long-form generation quality and speaker timbre consistency.

3.6 STREAMING INFERENCE

For streaming TTS, continuous generation can be re-anchored because the text input contains natural linguistic boundaries. We therefore apply a sentence-aware inference mechanism that discards the accumulated DiT positional state and preceding latents at each sentence boundary, then generate the next sentence independently from the speaker reference. A short overlap in the decoder output smooths the acoustic transition between consecutive sentences. This also reduces error accumulation of the target speaker timbre during long-form generation.

For streaming VC, we apply RAPR streaming to run generation as a continuous loop. For each incoming chunk, we first compute the 40 Hz ContentVec features, then assemble the RAPR context, generate the next chunk, decode it to a waveform and update the causal cache. Our main configuration uses 400 ms chunks, a 1 s history window, 4 NFEs, and no look-ahead. Together with the chunk-causal ContentVec encoder, this forms a fully causal VC pipeline.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Training and Evaluation Setup. CaDiS is trained on a large corpus of internal and publicly licensed speech data. The streaming TTS models are trained on 311k hours of speech data over three stages. The streaming VC models are trained on 179k hours in Stage 1 and 110k hours in each of Stage 2 and 3. We verify that no training data overlaps with the evaluation benchmarks described.

We assess zero-shot streaming TTS and cross-speaker streaming VC tasks on two benchmarks across normal human voices and expressive anime voices:

- Seed-TTS-eval (Anastassiou et al., 2024): the English `test-en` split of 1088 CommonVoice utterances for TTS and a non-parallel VC split of 1086 source-target utterance pairs.
- Anim400K (Cai et al., 2024): we select 1000 English anime clips to create 1000 TTS and VC pairs.

To support long-form evaluation in Section 4.6, we additionally construct 20 VC sources by concatenating consecutive utterances from the LibriSpeech `test-clean` set, yielding an average duration of 124 s. Each source is paired with a Seed-TTS-eval reference voice. We mainly report English evaluations. Mandarin and other results are provided along with the test-set details in appendices.

Model Setup. Both tasks use the same backbone with a separate model trained for each task. Following the three-stage training recipe, the final streaming model is the *Stage 3 DMD student*. The 0.8B-parameter offline bidirectional teacher is trained for 500k steps. The subsequent causal stages require 50k steps for the AR teacher on 32 A100 GPUs, 30k steps for the consistency student on 32 GPUs, and only 1000 steps for the DMD student on 64 GPUs. Additional data and hyperparameter details are provided in Appendix K.

Baselines. We generate outputs for all baselines from their released weights. For TTS, we compare against offline baselines F5-TTS, IndexTTS-2.5, and Fish-Speech S1-mini (Chen et al., 2025; Li et al., 2026; Liao et al., 2024); and the streaming baselines CosyVoice 3, Qwen3-TTS, VoXtream2, and Faster-IndexTTS-2 (Du et al., 2025; Hu et al., 2026; Torgashov et al., 2026; Du et al., 2026). For VC, we compare against the offline baseline Vevo (Zhang et al., 2025) and streaming baselines Seed-VC, X-VC, and MeanVC-2 (Liu, 2024; Zheng et al., 2026; Ma et al., 2026). Stage-wise ablation results are provided in Section 4.5, Appendices B and C.

Objective Metrics. We adopt word error rate (WER) (Radford et al., 2023), speaker similarity via WeSpeaker (Wang et al., 2023), and UTMOS (Saeki et al., 2022). SIM_{WeSp} uses the SimAM-ResNet100 checkpoint[1].

Subjective Metrics. We conduct subjective listening tests to evaluate models across both tasks. We use a gender-balanced subset of 200 clips per task and benchmark to let participants rate on a 1-5 scale

[1] https://github.com/wenet-e2e/wespeaker/blob/master/docs/pretrained.md

Table 1: Streaming TTS on Seed-TTS-eval English across objective and subjective metrics.

System	Stream	Objective			Subjective MOS		
		WER ↓	SIM_{WeSp} ↑	UTMOS ↑	NMOS ↑	QMOS ↑	SMOS ↑
Offline / Segment-level references							
Ours — offline	✗	1.75	0.816	4.29	3.90	4.17	3.85
F5-TTS	✗	1.91	0.815	4.15	3.78	4.01	3.79
IndexTTS-2.5	✗	2.58	0.791	3.98	3.62	3.80	3.68
S1-mini	✗	1.72	0.697	4.37	3.65	4.15	3.34
Streaming systems							
Qwen3-TTS	✓	**1.57**	0.769	**4.43**	3.91	4.09	3.69
VoXtream2	✓	1.98	0.803	3.98	3.62	3.92	3.78
CosyVoice3	✓	2.48	0.802	4.25	3.51	4.00	3.68
Faster-IndexTTS-2	✓	2.41	**0.813**	3.95	3.75	3.82	**3.80**
CaDiS (ours)	✓	1.69	0.785	4.19	**3.99**	**4.17**	3.79

on three questions: naturalness (NMOS), perceptual quality (QMOS), and speaker similarity (SMOS). For VC, we additionally ask participants to assess source-rhythm (RHY) and source-expression (EXP) preservation between the source and generated utterances. We collected around 400 submissions per TTS task (total 1200) by 200 participants and 300 submissions per VC task (total 900) by 100 participants. Full protocols and results are provided in Appendix J.

Streaming Metrics. To assess the streaming efficiency, we report first-packet latency (FPL) and real-time factor (RTF). The first-packet latency (FPL) of VC additionally includes the wall-clock compute required to produce the first output chunk, whereas TTS FPL is simply the wall-clock time to the first output packet. The RTF is computed as wall time/speech duration, where RTF$<$1 indicates real-time throughput. All timings are measured on an idle A100 GPU. Additional VC ablations on streaming assessments are reported in Appendices F–I.

4.2 STREAMING TTS RESULTS

Table 1 reports streaming TTS results on the Seed-TTS-eval English benchmark. Due to the page limit, the corresponding results on Anim400K are deferred to Table 15 in Appendix E.

Relative to our offline teacher, CaDiS retains comparable performance on both objective and subjective metrics using only 4 denoising steps. WER improves slightly from 1.75 to 1.69, while SIM_{WeSp} decreases from 0.816 to 0.785 and UTMOS from 4.29 to 4.19. The gap on subjective tests is smaller still, with a modest gain in naturalness and a slight reduction in speaker similarity. Overall, CaDiS largely preserves the perceptual quality of its offline counterpart.

Among the streaming baselines, CaDiS is particularly strong under human evaluation, achieving the best naturalness and audio quality scores, and the second-best speaker similarity score. On objective metrics, it achieves the second-lowest WER and the third-highest UTMOS score. While its speaker similarity trails the strongest baselines, it remains within a competitive range. Overall, these results indicate that CaDiS offers a favorable balance of intelligibility and human-rated perceptual quality.

4.3 STREAMING VOICE CONVERSION RESULTS

Table 2 reports the streaming VC results. Relative to our offline teacher, CaDiS improves both WER and speaker similarity (SIM_{WeSp}) but drops in UTMOS. The subjective results follow the same trend. However, among the streaming baselines, CaDiS achieves the far lowest WER, the highest UTMOS, and superior scores to nearly all subjective measures except SMOS. The main trade-off but also noteworthy observation is speaker similarity: X-VC achieves higher similarity (0.748 vs. 0.670), though the two systems are close in SMOS (3.16 vs. 3.10). Source preservation evaluations further reveal a complementary advantage of CaDiS. It achieves the highest rhythm (4.00) and expression (3.82) scores among the streaming baselines, closely matching its offline teacher.

We further evaluate VC performance on Anim400K in Table 3, whose expressive anime voices are entirely out-of-domain and span a broader range of timbres and styles. Among the streaming

Table 2: Streaming VC on Seed-TTS-eval English across objective and subjective metrics.

	Objective			Subjective MOS				
System	WER↓	SIM_{WeSp} ↑	UTMOS↑	NMOS↑	QMOS↑	SMOS↑	RHY↑	EXP↑
Offline references								
Ours — offline	3.11	0.653	4.11	3.92	3.87	3.01	3.95	3.81
Vevo	3.09	0.756	3.96	3.96	3.95	3.67	3.95	3.71
Streaming systems								
Seed-VC	5.85	<u>0.675</u>	3.67	3.32	2.99	2.90	3.65	3.43
X-VC	<u>3.41</u>	**0.748**	3.45	3.40	2.84	**3.16**	<u>3.85</u>	<u>3.64</u>
MeanVC-2	6.20	0.632	<u>3.79</u>	<u>3.58</u>	<u>3.26</u>	2.37	3.73	3.52
CaDiS	**3.09**	0.670	**3.90**	**3.79**	**3.73**	<u>3.10</u>	**4.00**	**3.82**

Table 3: Streaming VC on Anim400K (out-of-domain) across objective and subjective metrics.

		Objective			Subjective MOS		
System	Stream	WER ↓	SIM_{WeSp} ↑	UTMOS ↑	NMOS ↑	QMOS ↑	SMOS ↑
Offline references							
Ours — offline	✗	2.88	0.702	4.19	3.74	3.82	3.29
Vevo	✗	3.42	0.750	4.13	3.83	3.87	3.76
Streaming systems							
Seed-VC	✓	4.18	0.693	**4.09**	3.19	<u>3.64</u>	2.92
X-VC	✓	<u>3.45</u>	**0.741**	3.84	**3.60**	3.19	**3.28**
MeanVC-2	✓	5.64	0.655	3.95	3.28	3.20	2.59
CaDiS	✓	**3.11**	<u>0.695</u>	<u>4.04</u>	<u>3.58</u>	**3.73**	<u>3.12</u>

baselines, CaDiS again achieves the lowest WER, the highest QMOS, and second-best scores of all other metrics. CaDiS performs similarly to X-VC but yields substantially better perceptual quality in both UTMOS and QMOS.

Above all, while CaDiS does not maximize speaker similarity, it delivers the best intelligibility and perceptual quality, as well as preserves the best timing and expressive characteristics of source speech across in-domain and out-of-domain voices.

4.4 STREAMING EFFICIENCY

Table 4 further reports the streaming efficiency across both tasks. For TTS, CaDiS achieves 286 ms first-packet latency and 0.31 RTF, outperforming Qwen3-TTS and CosyVoice3. VoXtream2 has lower FPL but higher RTF, while Faster-IndexTTS-2 has lower RTF but a substantially higher FPL (553 ms). For VC, CaDiS trades a high FPL (575 ms) for zero-look-ahead operation while retaining the second-best RTF. MeanVC-2 exhibits stronger efficiency while its generation quality falls far below CaDiS and other baselines, making the efficiency gain a poor trade. X-VC, in contrast, incurs both a higher RTF and a less favorable combination of look-ahead and FPL than CaDiS, and does so while producing lower-quality speech.

Overall, these results show that CaDiS offers among the best overall streaming efficiency across baselines, and still delivers the best perceptual quality and intelligibility of speech generation.

4.5 TRAINING STAGE DECOMPOSITION

Table 5 decomposes the contribution of each training stage, and a consistent pattern emerges across both TTS and VC. Stage 1 introduces block causality, opening a modest quality gap to the offline teacher. Stage 2 reduces sampling steps from 32 to 4 at the cost of a further quality drop. Stage 3 DMD then recovers most of the quality loss at no extra inference cost. On TTS, Stage 2 trades WER and UTMOS, which are improved back by Stage 3, with the speaker similarity even improved on VC. Overall, the three stages play distinct and effective roles by enabling streaming capability, reducing inference cost, and restoring quality, respectively. They together form an integral pipeline of offline-to-streaming adaptation for a high-quality diffusion speech generator.

Table 4: First-packet latency (FPL), real-time factor (RTF), and look-ahead (LA) measured on A100.

(a) Streaming TTS

System	FPL (ms) ↓	RTF ↓
Qwen3-TTS	334	0.39
VoXtream2	**183**	0.32
CosyVoice3	1553	0.34
Faster-IndexTTS-2	553	**0.30**
CaDiS	286	0.31

(b) Streaming VC

System	LA (ms) ↓	FPL (ms) ↓	RTF ↓
Seed-VC	20	466	0.58
X-VC	100	285	0.35
MeanVC-2	40	**211**	**0.20**
CaDiS	**0**	575	0.26

Table 5: The ablation study of three training stages evaluated on Seed-TTS-eval English.

Stage	Steps	TTS WER ↓	TTS SIM_{WeSp} ↑	TTS UTMOS ↑	VC WER ↓	VC SIM_{WeSp} ↑	VC UTMOS ↑
Offline teacher (reference)	–	1.75	**0.816**	**4.29**	3.11	0.653	**4.11**
Stage 1: AR causal teacher	32	1.73	0.807	4.14	3.20	0.660	3.88
+Stage 2: consistency (CD)	4	2.05	0.767	3.82	3.23	0.639	3.76
+Stage 3: DMD (CaDiS)	4	**1.69**	0.785	4.19	**3.09**	**0.670**	3.90

4.6 Long-form streaming

We study long-form speaker identity control through VC, where the target speaker must be preserved continuously from a fixed reference. Figure 2(a) shows the effect of the RAPR history window when streaming long sequence. Without bounding the generated history, target-speaker similarity heavily degrades over time. A 10 s window reduces but does not eliminate this drift, while 0.5 s and 1 s windows keep identity nearly constant throughout the rollout. Thus, limiting generated history is critical for stable long-form VC. To understand why, we further obtain the decoder attention map assigned to the target prompt sink. For each generated query position, we sum its normalized attention weights over all keys that belong to the pinned target prompt, giving the fraction of attention the model assigns to the speaker reference. We then average this fraction across query positions, attention heads, decoder layers, and the four sampling steps to obtain one anchor-attention value for each generation chunk. As shown in Figure 2(b), short RAPR windows maintain a large, stable attention mass on the target reference, whereas longer or unbounded history progressively shifts attention away from the target. This trend closely tracks the speaker drift in Figure 2(a), indicating that bounded history preserves identity by sustaining attention to the reference. Overall, RAPR is essential for stable long-form VC. We use a 1 s window, which matches the long-form identity stability of 0.5 s while giving the best short-form WER (Appendix H).

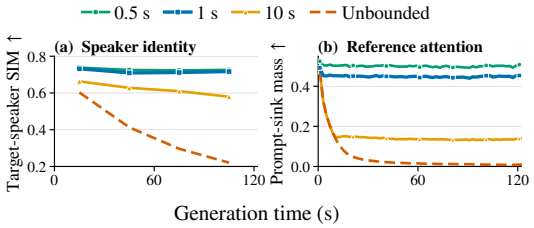


Figure 2: RAPR prevents speaker timbre drift in long-form streaming VC.

5 Conclusion

Streaming speech generation does not require abandoning high-quality diffusion generators in favor of autoregressive token models. We showed that a pretrained bidirectional speech Diffusion Transformer can be adapted into four-step generators through autoregressive diffusion adaptation, causal consistency distillation, and distribution matching refinement. The same backbone design supports two structurally different tasks: streaming TTS, where normalized positional encoding preserves text–speech alignment under block-causal generation, and streaming voice conversion, where frame-synchronous content conditioning and reference-anchored positional re-encoding maintain speaker identity under bounded context. Long-form experiments further show that RAPR limits speaker drift over extended streams by sustaining attention to the speaker reference. Across standard English speech and expressive anime speech benchmarks, the resulting streaming models approach their offline teachers while achieving competitive streaming efficiency in terms of first-packet latency and real-time factor. These results demonstrate the potential of high-quality diffusion speech generators for responsive and interactive streaming applications.

Reproducibility statement

We report the exact evaluation tooling (Whisper-large-v3, Paraformer-zh, WavLM-large-SV, Resemblyzer, SimAM-ResNet100, UTMOS) and run all scoring through the Seed-TTS-eval scripts so numbers are directly comparable to prior work; every baseline is generated in-house from released weights under the same scorer; and Section 4.1 specifies the distillation chain as a verified init-from sequence and names the GPU behind every latency, with warm-up discarded. Appendix K details the architecture, chunk/RoPE-cache and KV-cache configuration, per-task recipes, hyperparameters, and data mixes; Appendix A details test-set construction and the loudness/enhancement fairness protocol.

AI Use Statement

Generative AI tools were used to assist with research ideation and execution, drafting portions of the manuscript, language editing, and improving its readability. They were not used for literature retrieval or discovery, generating synthetic datasets, or proving mathematical claims. All AI-assisted ideas, analyses, and text were reviewed and verified by the authors, who take full responsibility for the content of the paper.

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, 2024. arXiv:2406.02430.

Kevin Cai, Chonghua Liu, and David M. Chan. ANIM-400K: A large-scale dataset for automated end-to-end dubbing of video. In *Proc. ICASSP*, pp. 11796–11800, 2024. doi: 10.1109/ICASSP48485.2024.10447911.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. In *Proc. NeurIPS*, volume 37, pp. 24081–24125, 2024a. doi: 10.52202/079017-0759.

Sanyuan Chen, Shujie Liu, Long Zhou, Yanqing Liu, Xu Tan, Jinyu Li, Sheng Zhao, Yao Qian, and Furu Wei. VALL-E 2: Neural codec language models are human parity zero-shot text to speech synthesizers, 2024b. arXiv:2406.05370.

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 *Proc. ACL*, pp. 6255–6271, 2025. doi: 10.18653/v1/2025.acl-long.313.

Ha-Yeong Choi and Sang-Hoon Lee. StreamFlow: Streaming audio generation from discrete tokens via streaming flow matching. In *Proc. NeurIPS*, volume 38, pp. 5243–5271, 2025. doi: 10.52202/085713-0165.

Trung Dang, David Aponte, Dung Tran, Tianyi Chen, and Kazuhito Koishida. Zero-shot text-to-speech from continuous text streams, 2024. arXiv:2410.00767.

Muyang Du, Shuang Yu, and Junjie Lai. Faster IndexTTS-2: Accelerating and streaming autoregressive zero-shot text-to-speech synthesis on GPUs, 2026. arXiv:2607.21042.

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, 2024. arXiv: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, 2025. arXiv:2505.17589.

Sefik Emre Eskimez, Xiaofei Wang, Manthan Thakker, Canrun Li, Chung-Hsien Tsai, Zhen Xiao, Hemin Yang, Zirun Zhu, Min Tang, Xu Tan, Yanqing Liu, Sheng Zhao, and Naoyuki Kanda. E2 TTS: Embarrassingly easy fully non-autoregressive zero-shot TTS. In *Proc. SLT*, pp. 682–689, 2024. doi: 10.1109/SLT61566.2024.10832320.

Dake Guo, Jixun Yao, Lihan Ma, He Wang, and Lei Xie. StreamFlow: Streaming flow matching with block-wise guided attention mask for speech token decoding. In *Proc. NCMMSC*, Communications in Computer and Information Science, pp. 75–86. Springer Nature Singapore, 2026. doi: 10.1007/978-981-95-5382-2_7.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. arXiv: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, 2026. arXiv:2601.15621.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. In *Proc. NeurIPS*, volume 38, pp. 185411–185436, 2025. doi: 10.52202/085713-5576.

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 *Proc. ICML*, volume 267, pp. 27255–27270. PMLR, 2025.

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 *Proc. ICML*, volume 235, pp. 22605–22623. PMLR, 2024.

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 *Proc. NeurIPS*, volume 36, pp. 14005–14034, 2023. doi: 10.52202/075280-0618.

Yinghao Aaron Li, Rithesh Kumar, and Zeyu Jin. DMOSpeech: Direct metric optimization via distilled diffusion model in zero-shot speech synthesis. In *Proc. ICML*, volume 267, pp. 35186–35208. PMLR, 2025.

Yunpei Li, Xun Zhou, Jinchao Wang, Lu Wang, Yong Wu, Siyi Zhou, Yiquan Zhou, and Jingchen Shu. IndexTTS 2.5 technical report, 2026. arXiv:2601.03888.

Shijia Liao, Yuxuan Wang, Tianyu Li, Yifan Cheng, Ruoyi Zhang, Rongzhi Zhou, and Yijin Xing. Fish-Speech: Leveraging large language models for advanced multilingual text-to-speech synthesis, 2024. arXiv:2411.01156.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *Proc. ICLR*, 2023.

Songting Liu. Zero-shot voice conversion with diffusion transformers, 2024. arXiv:2411.09943.

Yanqing Liu, Ruiqing Xue, Chong Zhang, Yufei Liu, Gang Wang, Bohan Li, Yao Qian, Lei He, Shujie Liu, and Sheng Zhao. Next tokens denoising for speech synthesis, 2025. arXiv:2507.22746.

Zhijun Liu, Shuai Wang, Sho Inoue, Qibing Bai, and Haizhou Li. Autoregressive diffusion transformer for text-to-speech synthesis, 2024. arXiv:2406.05551.

Guobin Ma, Yuxuan Xia, Yuepeng Jiang, Dake Guo, Hanke Xie, Jingbin Hu, Yanbo Wang, Lei Xie, and Pengcheng Zhu. MeanVC 2: Robust low-latency streaming zero-shot voice conversion. In *Proc. Interspeech*, pp. 4721–4725, 2026. doi: 10.21437/Interspeech.2026-1961.

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 *Proc. ACL*, pp. 1287–1300, 2025. doi: 10.18653/v1/2025.acl-long.65.

Ziqian Ning, Yuepeng Jiang, Pengcheng Zhu, Jixun Yao, Shuai Wang, Lei Xie, and Mengxiao Bi. DualVC: Dual-mode voice conversion using intra-model knowledge distillation and hybrid predictive coding. In *Proc. Interspeech*, pp. 2063–2067, 2023. doi: 10.21437/Interspeech.2023-1157.

Ziqian Ning, Shuai Wang, Pengcheng Zhu, Zhichao Wang, Jixun Yao, Lei Xie, and Mengxiao Bi. DualVC 3: Leveraging language model generated pseudo context for end-to-end low latency streaming voice conversion. In *Proc. Interspeech*, pp. 197–201, 2024. doi: 10.21437/Interspeech.2024-1857.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proc. ICCV*, pp. 4195–4205, 2023. doi: 10.1109/ICCV51070.2023.00387.

Kaizhi Qian, Yang Zhang, Heting Gao, Junrui Ni, Cheng-I Lai, David Cox, Mark Hasegawa-Johnson, and Shiyu Chang. ContentVec: An improved self-supervised speech representation by disentangling speakers. In *Proc. ICML*, volume 162, pp. 18003–18017. PMLR, 2022.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *Proc. ICML*, volume 202, pp. 28492–28518. PMLR, 2023.

David Ruhe, Jonathan Heek, Tim Salimans, and Emiel Hoogeboom. Rolling diffusion models. In *Proc. ICML*, volume 235, pp. 42818–42835. PMLR, 2024.

Takaaki Saeki, Detai Xin, Wataru Nakata, Tomoki Koriyama, Shinnosuke Takamichi, and Hiroshi Saruwatari. UTMOS: UTokyo-SaruLab system for VoiceMOS challenge 2022. In *Proc. Interspeech*, pp. 4521–4525, 2022. doi: 10.21437/Interspeech.2022-439.

Prem Seetharaman and Rithesh Kumar. Taming audio VAEs via target-KL regularization. In *Proc. ICASSP*, pp. 15487–15491, 2026. doi: 10.1109/ICASSP55912.2026.11460662.

Zhengyan Sheng, Zhihao Du, Shiliang Zhang, Zhijie Yan, and Liping Chen. SyncSpeech: Efficient and low-latency text-to-speech based on temporal masked transformer. In *Proc. ICASSP*, pp. 16682–16686, 2026. doi: 10.1109/ICASSP55912.2026.11460607.

ShoukanLabs. AniSpeech dataset. Hugging Face dataset, 2023. URL `https://huggingface.co/datasets/ShoukanLabs/AniSpeech`. Accessed September 25, 2026.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proc. ICML*, volume 202, pp. 32211–32252. PMLR, 2023.

Zhiyao Sun, Ziqiao Peng, Yifeng Ma, Yi Chen, Zhengguang Zhou, Zixiang Zhou, Guozhen Zhang, Youliang Zhang, Yuan Zhou, Qinglin Lu, and Yong-Jin Liu. StreamAvatar: Streaming diffusion models for real-time interactive human avatars. In *Proc. CVPR*, pp. 10887–10897, 2026.

Nikita Torgashov, Gustav Eje Henter, and Gabriel Skantze. VoXtream2: Full-stream TTS with dynamic speaking rate control, 2026. arXiv:2603.13518.

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 *Proc. ICASSP*, pp. 1–5, 2023. doi: 10.1109/ICASSP49357.2023.10096626.

Zhichao Wang, Yuanzhe Chen, Xinsheng Wang, Lei Xie, and Yuping Wang. StreamVoice: Streamable context-aware language modeling for real-time zero-shot voice conversion. In *Proc. ACL*, pp. 7328–7338, 2024. doi: 10.18653/v1/2024.acl-long.396.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proc. NAACL-HLT*, pp. 483–498, 2021.

Yang Yang, Yury Kartynnik, Yunpeng Li, Jiuqiang Tang, Xing Li, George Sung, and Matthias Grundmann. StreamVC: Real-time low-latency voice conversion. In *Proc. ICASSP*, pp. 11016–11020, 2024. doi: 10.1109/ICASSP48485.2024.10446863.

Zhen Ye, Wei Xue, Xu Tan, Jie Chen, Qifeng Liu, and Yike Guo. CoMoSpeech: One-step speech and singing voice synthesis via consistency model. In *Proc. ACM Multimed.*, pp. 1831–1839, 2023. doi: 10.1145/3581783.3612061.

Zhen Ye, Zeqian Ju, Haohe Liu, Xu Tan, Jianyi Chen, Yiwen Lu, Peiwen Sun, Jiahao Pan, Weizhen Bian, Shulin He, Wei Xue, Qifeng Liu, and Yike Guo. FlashSpeech: Efficient zero-shot speech synthesis. In *Proc. ACM Multimed.*, pp. 6998–7007, 2024. doi: 10.1145/3664647.3681044.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Frédo Durand, and William T. Freeman. Improved distribution matching distillation for fast image synthesis. In *Proc. NeurIPS*, volume 37, pp. 47455–47487, 2024a. doi: 10.52202/079017-1505.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proc. CVPR*, pp. 6613–6623, 2024b.

Tianwei Yin, Qiang Zhang, Richard Zhang, William T. Freeman, Frédo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. In *Proc. CVPR*, pp. 22963–22974, 2025.

Xueyao Zhang, Xiaohui Zhang, Kainan Peng, Zhenyu Tang, Vimal Manohar, Yingru Liu, Jeff Hwang, Dangna Li, Yuhao Wang, Julian Chan, Yuan Huang, Zhizheng Wu, and Mingbo Ma. Vevo: Controllable zero-shot voice imitation with self-supervised disentanglement. In *Proc. ICLR*, 2025.

Min Zhao, Hongzhou Zhu, Kaiwen Zheng, Zihan Zhou, Bokai Yan, Xinyuan Li, Xiao Yang, Chongxuan Li, and Jun Zhu. Causal forcing++: Scalable few-step autoregressive diffusion distillation for real-time interactive video generation, 2026. arXiv:2605.15141.

Qixi Zheng, Yuxiang Zhao, Tianrui Wang, Wenxi Chen, Kele Xu, Yikang Li, Qinyuan Cheng, Xipeng Qiu, Kai Yu, and Xie Chen. X-VC: Zero-shot streaming voice conversion in codec space, 2026. arXiv:2604.12456.

A TEST-SET CONSTRUCTION AND FAIRNESS PROTOCOL

All short-form results use the public Seed-TTS-eval benchmark (Anastassiou et al., 2024), which ships a native zero-shot VC task—its released `non_para_reconstruct` split (a source utterance + a distinct non-parallel speaker as timbre reference), used verbatim—scored with the same WER/CER/SIM tooling as its TTS task, so TTS and VC are directly comparable. WER uses Whisper-large-v3 and CER uses Paraformer-zh (FunASR), both via the Seed-TTS-eval scripts verbatim; SIM_{Res} is Resemblyzer cosine, $\text{SIM}_{\text{WavLM}}$ is WavLM-large-SV cosine, and SIM_{WeSp} is SimAM-ResNet100/VoxBlink2 cosine. Every system converts toward, and is scored against, the *same* reference voice pre-processed by an enhancement front end. For Seed-TTS-eval the source/content audio is left unmodified; the anime out-of-domain pools (Appendix E) also enhance the source, so both source and target are enhanced there. Streaming and offline audio are loudness-normalized before scoring, and all objective metrics used here are approximately level-invariant. Long-form TTS uses 100 English passages ($\sim$48 s) and Mandarin sentence-concatenations ($\sim$145 s). For the English set, we stitch disjoint, already-punctuated Seed-TTS-eval sentences into the 100 passages. Long-form VC uses LibriSpeech-Long sources ($\sim$123 s, English) and AISHELL-3 single-speaker sources ($\sim$122 s, Mandarin). For English VC, we stitch consecutive utterances read within the same LibriSpeech `test-clean` chapter into 20 sources averaging $\sim$124 s, then pair each source with a Seed-TTS-eval reference voice so that the long- and short-form sets have the same target-speaker distribution.

Our primary anime benchmark is **Anim400K** (Cai et al., 2024), approximately 425K aligned animated video segments in Japanese and English collected from TV anime episodes; we use only its English audio and apply an audio-quality, single-speaker-purity, and caption-faithfulness gate to select 1000 clips, from which we construct 1000 TTS pairs and 1000 VC pairs. We additionally include **AniSpeech** (ShoukanLabs, 2023) (23,656 clips from 265 voices) as a second anime benchmark for breadth and likewise construct 1000 TTS pairs and 1000 VC pairs.

Table 6: **Short-form zero-shot TTS, Seed-TTS-eval English — full table** (enhanced targets). Companion to main-text Table 1 with the Resemblyzer (SIM_{Res}) and WavLM-large-SV ($\text{SIM}_{\text{WavLM}}$) similarity variants (the main text reports WeSpeaker), plus the AR-teacher and CD rows.

System	Stream	WER ↓	SIM_{Res} ↑	$\text{SIM}_{\text{WavLM}}$ ↑	UTMOS ↑	FPL (ms) ↓	RTF ↓
Offline references							
Ours — offline	✗	1.75	0.910	0.677	4.29	–	–
Qwen3-TTS (offline)	✗	1.52	0.886	0.670	4.43	–	–
F5-TTS	✗	1.91	0.897	0.657	4.15	–	–
Faster-IndexTTS-2 (offline)	✗	2.04	0.898	0.665	4.04	–	–
CosyVoice3 (offline)	✗	1.91	0.906	0.697	4.26	–	–
IndexTTS-2.5	✗	2.58	0.904	0.607	3.98	–	–
S1-mini	✗	1.72	0.861	0.579	4.37	–	–
Streaming systems							
Qwen3-TTS	✓	**1.57**	0.880	0.659	**4.43**	334	0.39
VoXtream2	✓	1.98	0.896	0.695	3.98	**183**	0.32
CosyVoice3	✓	2.48	**0.907**	0.695	4.25	1553	0.34
Faster-IndexTTS-2	✓	2.41	0.900	0.670	3.95	553	**0.30**
Ours — AR teacher (32-step)	✗	1.73	0.901	0.665	4.14	-	-
Ours — CD (4-step)	✓	2.05	0.887	0.631	3.82	286	0.31
Ours — DMD (4-step)	✓	1.69	0.897	0.647	4.19	286	0.31

B ENGLISH OBJECTIVE RESULTS WITH BOTH SIMILARITY METRICS

Both main-text short-form English tables (Tables 1 and 2) headline WeSpeaker (SimAM-ResNet100/VoxBlink2), our primary speaker-similarity metric. For completeness we repeat both tables here with the two secondary variants, Resemblyzer (SIM_{Res}) and WavLM-large-SV ($\text{SIM}_{\text{WavLM}}$); every other value is identical to the main text. The metrics rank speaker similarity differently—most visibly on VC, where WavLM-SV places MeanVC-2 highest among streaming systems while WeS-

Table 7: **Short-form zero-shot streaming VC, Seed-TTS-eval English — full table**. Companion to main-text Table 2 with all three similarity metrics (WeSpeaker, Resemblyzer, WavLM-large-SV), plus the internal AR-teacher and CD rows.

System	Stream	WER ↓	SIM_{WeSp} ↑	SIM_{Res} ↑	SIM_{WavLM} ↑	UTMOS ↑	FPL (ms) ↓	RTF ↓
Offline references								
Ours — offline	✗	3.11	0.653	0.848	0.476	**4.11**	–	–
Vevo	✗	3.09	**0.756**	**0.881**	0.553	3.96	–	–
Streaming systems								
Seed-VC	✓	5.85	0.675	0.868	0.439	3.67	466	0.58
X-VC	✓	3.41	**0.748**	**0.886**	0.604	3.45	285	0.35
MeanVC-2	✓	6.20	0.632	0.824	**0.616**	3.79	**211**	**0.20**
Ours — AR teacher (32-step)	✗	3.20	0.660	0.830	0.464	3.88	-	-
Ours — CD (4-step)	✓	3.23	0.639	0.829	0.447	3.76	575	0.26
Ours — DMD (4-step)	✓	**3.09**	0.670	0.842	0.465	**3.90**	575	0.26

peaker and the listening study both place it lowest, and where the gap from our streaming student to the X-VC leader is far smaller on WeSpeaker than on WavLM-SV. Both full tables also include our intermediate-stage ablations—the 32-step AR causal teacher and the CD student. The AR teacher is chunk-causal but, at 32 steps, slower than real time, so we mark it non-streaming and report it only as an internal quality reference rather than a competing real-time system; the CD student shares the DMD student's architecture, and therefore the same first-packet latency and RTF.

Table 8: Short-form zero-shot TTS, Seed-TTS-eval Mandarin. CER via Paraformer-zh; SIM_{Res} headline.

System	Stream	CER ↓	SIM_{Res} ↑	UTMOS ↑
Offline references				
Ours — offline	✗	1.76	0.895	3.42
F5-TTS	✗	1.54	0.906	3.34
Faster-IndexTTS-2(offline)	✗	0.93	0.899	3.34
IndexTTS-2.5	✗	2.02	0.901	3.32
S1-mini	✗	1.01	0.850	3.82
Streaming systems				
Qwen3-TTS	✓	**0.91**	0.881	**3.94**
CosyVoice3	✓	1.41	**0.910**	3.65
Faster-IndexTTS-2	✓	1.26	0.901	3.15
Ours — AR teacher (32-step)	✗	1.87	0.880	3.33
Ours — CD (4-step)	✓	2.77	0.872	2.97
Ours — DMD (4-step)	✓	1.70	0.887	3.41

Table 9: Short-form zero-shot streaming VC, Seed-TTS-eval Mandarin. Latency/RTF are the arch-shared values reported in Table 4. Our CER is competitive with X-VC; the WavLM-SV gap to baselines narrows relative to English.

System	Stream	CER ↓	SIM_{Res} ↑	UTMOS ↑
Offline references				
Ours — offline	✗	2.82	0.854	3.36
Streaming systems				
Seed-VC	✓	4.28	0.838	**3.43**
X-VC	✓	**2.73**	**0.894**	2.49
MeanVC-2	✓	4.08	0.836	3.15
Ours — AR teacher (32-step)	✗	3.09	0.838	3.03
Ours — CD (4-step)	✓	3.29	0.833	2.91
Ours — DMD (4-step)	✓	3.26	0.841	3.06

C SHORT-FORM MANDARIN RESULTS

Tables 8 and 9 report short-form Mandarin. On TTS the offline→streaming delta is now negligible in both languages—the streaming DMD student even edges the offline model on CER/WER (Mandarin 1.70 vs. 1.76, English 1.69 vs. 1.75)—while VC incurs a modest $+0.44$ CER cost. UTMOS is English-trained, so its Mandarin values are for cross-system ranking only. On TTS, our streaming DMD is mid-pack on CER at competitive similarity; the AR causal teacher is again the stronger-but-slower point (causal but not real-time, so we mark it non-streaming) and consistency distillation the weak link, matching the English ablation.

D LONG-FORM RESULTS

The long-form benchmarks are designed to evaluate both output quality and speaker-similarity drift during extended generation.

Table 10 gives English long-form *TTS* identity consistency (windowed Resemblyzer similarity: whole-clip SIM, early→late halves, absolute drift, slope per minute, and long-form WER); long-form *VC* is carried by Figure 2 in the main text, with the full Mandarin table in Table 12 below. TTS re-anchors per sentence and is flat by construction.

Table 10: Long-form TTS identity consistency, English (100 passages ∼48 s).

System	whole-SIM$_{\text{Res}}$ ↑	early→late	drift ↓	slope/min ↑	long-WER ↓
CaDiS	0.935	0.917→0.909	0.009	**−0.031**	6.81
CosyVoice3	**0.938**	0.924→0.916	0.008	−0.034	7.24
Qwen3-TTS	0.885	0.875→0.855	0.021	−0.122	**4.93**

Table 11: Long-form TTS, Mandarin (sentence-concat ∼145 s, n=20, enhanced targets). CER as %.

System	Stream	whole-SIM$_{\text{Res}}$ ↑	early→late	drift ↓	slope/min ↑	long-CER ↓
CaDiS	✓	0.922	0.899→0.891	**0.008**	**−0.004**	8.4
CosyVoice3	✓	**0.930**	0.906→0.894	0.013	−0.006	7.4
S1-mini	✗	0.879	0.853→0.844	0.010	−0.006	4.8
Qwen3-TTS	✓	0.864	0.873→0.794	0.079	−0.056	**4.6**

Table 12: Long-form VC, Mandarin (AISHELL-3 single-speaker src ∼122 s, n=20, enhanced targets). CER as %.

System	Stream	whole-SIM$_{\text{Res}}$ ↑	early→late	drift ↓	slope/min ↑	long-CER ↓
Offline references						
Ours — offline	✗	0.793	0.775→0.769	0.006	−0.004	7.2
Streaming systems						
CaDiS	✓	0.869	0.837→0.814	0.023	−0.016	8.7
X-VC	✓	**0.921**	0.903→0.891	0.012	−0.009	**7.1**
MeanVC-2	✓	0.872	0.851→0.841	0.010	−0.009	9.5
Seed-VC	✓	0.752	0.726→0.722	**0.004**	−0.015	24.4

Tables 11 and 12 give Mandarin long-form drift. As in English, our TTS keeps the flattest slope at a bounded window, and VC drifts gracefully with content held by the live source.

E OUT-OF-DOMAIN ANIME RESULTS

Anim400K is our **main anime benchmark**: its cross-speaker VC results are promoted to the main text (Table 3) and its TTS results appear in Table 15. We additionally report **AniSpeech** (Tables 13–14)

Table 13: TTS on AniSpeech. [a]CosyVoice3 streams but is not interactively real-time at our 400 ms regime (FPL 1553 ms, LLM-first-hop bound).

System	Stream	WER ↓	SIM_{WeSp} ↑	$\textit{SIM}_{\text{WavLM}}$ ↑	UTMOS ↑	FPL (ms) ↓	RTF ↓
Offline references							
Ours — offline	✗	**2.64**	0.740	*0.626*	3.89	–	–
F5-TTS	✗	3.34	0.748	***0.628***	3.74	–	–
IndexTTS-2.5	✗	5.31	**0.750**	*0.588*	3.67	–	–
S1-mini	✗	3.07	0.604	*0.548*	**4.23**	–	–
Streaming systems							
CaDiS (4-step)	✓	3.12	0.690	*0.590*	3.67	286	0.31
Qwen3-TTS	✓	**2.65**	0.668	*0.601*	3.98	334	0.39
VoXtream2	✓	5.24	0.694	*0.631*	3.37	**183**	0.32
CosyVoice3[a]	✓	3.06	**0.771**	***0.645***	**4.04**	1553	0.34
Faster-IndexTTS-2	✓	5.36	0.761	*0.638*	3.51	553	**0.30**

Table 14: Streaming cross-speaker VC on AniSpeech (additional anime benchmark, OOD; enhanced targets). The two *Ours* rows isolate the direct-metric addition (Appendix F). [a]MeanVC-2 is Mandarin-only (English is OOD for it).

System	Stream	WER ↓	SIM_{WeSp} ↑	$\textit{SIM}_{\text{WavLM}}$ ↑	UTMOS ↑	FPL (ms) ↓	RTF ↓
Offline references							
Ours — offline	✗	5.58	0.596	*0.408*	**3.47**	–	–
Streaming systems							
Seed-VC	✓	15.82	0.663	*0.440*	3.06	466	0.58
X-VC	✓	**4.92**	**0.674**	*0.525*	2.88	285	0.35
MeanVC-2[a]	✓	9.04	0.596	***0.588***	3.46	**211**	**0.20**
Ours — DMD	✓	5.30	0.576	*0.399*	3.37	575	0.26
Ours — DMD (+DMOSpeech)	✓	5.15	0.571	*0.395*	3.41	575	0.26

as a second anime benchmark for breadth. On both, SIM_{WeSp} (SimAM-ResNet100/VoxBlink2) is the headline speaker metric, and both anime source and target are enhanced. Before constructing the evaluation pairs, we apply model-independent audio-quality, single-speaker-purity, and transcript-consistency filtering. The transcript-consistency check removes segments containing approximately 1 s of neighboring speech caused by imperfect utterance boundaries. Filtering is completed before system evaluation, and no scoring-time trimming is applied.

F VC DIRECT-METRIC ABLATION

The two *Ours* VC rows in Table 14 isolate the effect of adding DMOSpeech-style (Li et al., 2025) direct-metric content/speaker losses on top of the rescale-only student, which we take as the comparison base. The direct-metric term is flat on raw speaker cosine (WeSpeaker $0.576 \rightarrow 0.571$, within noise) but improves intelligibility and naturalness (WER $5.30 \rightarrow 5.15$, UTMOS $3.37 \rightarrow 3.41$). In other words, the direct-metric term buys WER/UTMOS, not speaker cosine.

G VC CONTENT-REPRESENTATION ABLATION: NATIVE 40 Hz VS. 50 Hz

Matching the content rate to the acoustic-latent rate is a design choice, not just an implementation convenience. To separate the effect of the *rate* from the effect of making the encoder *causal and streamable*, we probe three ContentVec encoders *intrinsically*—without retraining the VC system—on LibriSpeech test-clean (40 speakers $\times$ 12 utterances): the pretrained 50 Hz teacher (bidirectional reference); a *strided* 40 Hz encoder that only relocates the convolutional stride to emit 40 Hz (weights copied from the teacher, no training, still bidirectional); and the deployed *native* 40 Hz encoder, distilled from the teacher with causal convolutions and block-causal self-attention. We measure

Table 15: **TTS on Anim400K**. Objective metrics on the full benchmark plus subjective MOS from a gender-balanced listening study (Prolific; protocol in Appendix J). All systems in the table were evaluated together in the same collection. Streaming−offline (Ours): $\Delta_{\text{NMOS}}=-0.11$, $\Delta_{\text{QMOS}}=-0.23$, $\Delta_{\text{SMOS}}=-0.21$. The 95% CIs for QMOS and SMOS exclude 0, whereas the NMOS CI includes 0 ($[-0.22, +0.00]$); on this out-of-domain benchmark the streaming student does not fully close the gap to the offline teacher.

		Objective			Subjective MOS		
System	Stream	WER ↓	SIM_{WeSp} ↑	UTMOS ↑	NMOS ↑	QMOS ↑	SMOS ↑
Offline references							
Ours — offline	✗	**1.14**	**0.788**	4.32	**3.66**	**4.10**	**3.54**
F5-TTS	✗	1.79	0.781	4.21	3.41	3.91	3.28
IndexTTS-2.5	✗	3.21	0.765	4.05	3.51	3.89	3.39
S1-mini	✗	1.16	0.675	**4.41**	3.36	4.02	3.10
Streaming systems							
Qwen3-TTS	✓	**0.90**	0.712	4.23	**3.73**	**4.08**	3.24
VoXtream2	✓	1.51	0.762	4.12	3.39	3.91	3.35
CosyVoice3	✓	1.39	**0.784**	**4.33**	3.60	**4.08**	**3.47**
Faster-IndexTTS-2	✓	3.29	0.773	3.99	3.42	3.73	3.26
CaDiS (4-step)	✓	1.17	0.753	4.20	3.55	3.87	3.33

content (phone error rate from a shared rate-agnostic CTC probe, so 40 vs. 50 Hz are comparable), speaker leakage (layer-12 linear speaker-classification accuracy; lower is more invariant), pitch leakage (ridge F0 R^2; lower is more invariant), and cosine to the teacher representation (Table 16).

The decomposition is clean. Changing the rate alone (teacher50→strided40) is nearly free on content (+0.22 pt PER) and leaves speaker and pitch leakage flat or slightly lower; the larger cost (+1.86 pt PER, teacher-cos $0.911 \rightarrow 0.847$) comes from making the encoder causal and streamable, not from the rate change. The native 40 Hz representation thus preserves content and disentanglement while removing the condition↔latent rate mismatch and enabling direct additive conditioning—confirming that the 40 Hz design is functionally motivated rather than a convenience.

Table 16: **Intrinsic content-representation probes** for the ContentVec encoder (source-rate, no VC retraining; LibriSpeech test-clean, 40 spk × 12 utt). PER from a shared rate-agnostic CTC probe (content); spk-acc is layer-12 linear speaker classification (leakage, chance 0.025); F0 R^2 is ridge pitch prediction (leakage); teacher-cos is cosine to the 50 Hz teacher. *teacher50→strided40* isolates the pure rate change; *strided40→native40* isolates the causal/streaming-distillation cost.

Encoder	PER ↓	spk-acc ↓	F0 R^2 ↓	teacher-cos ↑
teacher50 (50 Hz bidirectional, reference)	**5.39**	0.690	0.356	1.000
strided40 (40 Hz, no training)	5.61	**0.661**	**0.346**	0.911
native40 (distilled causal, deployed)	7.47	0.674	0.358	0.847

H RAPR WINDOW ABLATION: SHORT-FORM QUALITY, LONG-FORM DRIFT, AND ATTENTION

We ablate reference-anchored positional re-encoding by training four DMD students that differ *only* in the rolling clean-history window (train=test, 0.5/1/4/10 s; the ∼10 s speaker reference and everything else are held fixed), plus an unbounded (RAPR-off) arm evaluated on the 1 s model. These two ablation tables, with the long-form sweep in the main text (Figure 2(a)), tell one story—a *short* window is best.

Short-form quality (Table 17). On both English and Mandarin Seed-TTS-eval VC, speaker similarity (Resemblyzer and WeSpeaker) falls *monotonically* as the window grows, while intelligibility is flat (EN) or improves only $\sim$0.4 pt (ZH); the 1 s row reproduces the deployed anchor. The only thing a longer window buys is a small ZH content gain, at a similarity and cost penalty.

Long-form drift (main text, Figure 2(a); full Mandarin table in Table 12). Over minute-long rollouts, bounded windows all hold and *smaller bounds drift less, monotonically*, while disabling RAPR (unbounded) collapses identity. Bounding is essential and a short bound is best—the opposite of the "longer context helps" intuition.

Mechanism (Figure 2(b), Table 18). Recovering the decoder self-attention over the same 20 rollouts shows the pinned prompt-sink's share of attention falls monotonically with window size ($0.467 \rightarrow 0.121 \rightarrow 0.011$) and tracks the drift ranking one-for-one: the RAPR window sets how large a fraction of the KV cache the fixed speaker anchor occupies. A short window lets it dominate; an unbounded window lets accumulated (and drifting) source-history wash it out. This is the mechanism behind the VC identity claim, measured rather than argued.

Table 17: **RAPR rolling-window sweep, short-form Seed-TTS-eval VC** (4 steps, enhanced sources; speaker reference fixed at $\sim$10 s; only the train-matched rolling window varies). Speaker similarity (Res=Resemblyzer headline, WeSp=WeSpeaker) falls monotonically as the window grows on both languages; the 1 s row ($\star$, deployed) reproduces the anchor exactly.

	English				Mandarin			
Window	WER ↓	SIM_{Res} ↑	SIM_{WeSp} ↑	UTMOS	CER ↓	SIM_{Res} ↑	SIM_{WeSp} ↑	UTMOS
0.5 s	3.33	**0.849**	**0.681**	3.90	3.45	**0.847**	**0.776**	3.07
1 s*	**3.09**	0.842	0.670	3.90	3.26	0.841	0.772	3.06
4 s	3.16	0.834	0.657	3.88	3.05	0.828	0.758	3.06
10 s	3.16	0.835	0.657	3.88	**3.01**	0.828	0.757	3.07

Table 18: **RAPR attention mechanism** (decoder self-attention recovered from 20 cross-speaker rollouts $\sim$121 s, averaged over steps/heads/queries/20 layers and clips; late half $t \geq t_{\text{last}}/2$). The three key regions partition the attention mass (rows sum to 1). Mass on the pinned prompt-sink falls monotonically as the window grows—displaced by the accumulating clean-history region—and tracks the long-form drift ranking (Figure 2(a)) one-for-one.

Window	prompt-sink ↑	clean-history	current
0.5 s	**0.467**	0.160	0.373
1 s*	0.417	0.226	0.356
10 s	0.121	0.594	0.286
Unbounded	0.011	0.755	0.234

I SAMPLING-STEP BUDGET

We sweep student NFEs ($1/2/4$ vs. the 32-step teacher). Four steps is the knee of the quality–latency curve: TTS intelligibility and naturalness reach the offline ceiling while RTF stays at 0.31; below four steps UTMOS degrades faster than WER, and speaker similarity is the last to fall. For streaming VC, Table 19 sweeps $1/2/4/8$ steps on the deployed VC DMD student (EN Seed-TTS-eval; everything else fixed). The same knee appears: WER and speaker similarity (Resemblyzer) both peak at four steps and UTMOS reaches within 0.05 of its ceiling, while going to eight steps *regresses* WER ($3.09 \rightarrow 3.24$) and Resemblyzer ($0.842 \rightarrow 0.838$)—over-denoising—for a doubled per-chunk cost. One step is fastest but sacrifices naturalness (UTMOS 3.33).

J SUBJECTIVE EVALUATION PROTOCOL

Because neural MOS predictors saturate on strong systems, we validate naturalness with a human listening study. Rather than one omnibus questionnaire, we decompose each evaluation into *separate*

Table 19: Streaming VC sampling-step budget on the deployed VC DMD student (EN Seed-TTS-eval, enhanced sources; ∼1 s rolling context, ∼10 s reference). Only the per-chunk step count varies; the four-step row reproduces the deployed system exactly. SIM_{Res} (Resemblyzer) is the headline and SIM_{WeSp} (WeSpeaker) secondary. Four steps (⋆) is the deployed default; deployed latency is reported in Table 4.

Steps	WER ↓	SIM_{Res} ↑	SIM_{WeSp} ↑	UTMOS ↑
1	3.57	0.831	0.629	3.33
2	3.23	0.841	0.654	3.76
4 ⋆	**3.09**	**0.842**	0.670	3.90
8	3.24	0.838	**0.672**	**3.95**

rating tasks—**two** for TTS and **three** for VC—each a self-contained Prolific instrument with its own instructions, its own on-screen reference condition, and its own randomized trial pages. Within a task, all systems for that task–language–benchmark are rated together and a given rater/item pair never crosses tasks. The two shared tasks are (i) *naturalness and quality*, which scores the synthesized recording *in isolation* on five-point naturalness (NMOS) and quality/artifact (QMOS) scales, and (ii) *target-speaker similarity* (SMOS), which plays the recording beside a reference timbre—the prompt for TTS, the target for VC. VC adds a third task, (iii) *source preservation*, which plays the converted output beside the *source* recording and scores rhythm/timing (RHY) and expression/emotion (EXP); lexical content is already covered by objective WER. Every rating is a five-point absolute score. Raters are Prolific English speakers (English evaluations) or native/fluent-Mandarin speakers (Mandarin evaluations), item-blocked with two votes per cell; attention-failed assignments are dropped whole (6–18% per English family, 17–28% per Mandarin family), means carry participant-cluster bootstrap 95% CIs. Listening sets are gender-balanced, specifically 100 male utterances and 100 female utterances for TTS experiments, and 50 male-to-female conversion, 50 female-to-male conversion, 50 male-to-male conversion, and 50 female-to-female conversion for VC experiments. The results are reported as follows: the English Seed-TTS-eval results are in the main text (TTS Table 1, Section 4.2; VC Table 2, Section 4.3); the out-of-domain Anim400K results are folded into Appendix E (TTS Table 15, VC Table 20); and the Mandarin (Seed-TTS-eval) results are reported below (Tables 21–22).

Rating instruments and reference conditions. Figures 3–5 show the English interface for every task. The *naturalness and quality* task (Fig. 4a for TTS, Fig. 5a for VC) presents a single unlabeled recording and asks two questions—"How natural does the speech sound overall—like a real human speaker, flowing smoothly, without choppy or halting stretches?" (NMOS) and "Is the speech clear and free of audio artifacts?" (QMOS)—each on a 1 (Very poor) to 5 (Excellent) scale with an N/A option. Crucially, *no reference audio is shown*: the rater hears only the system output. The *similarity* task (Fig. 4b, Fig. 5b) is the only task that reveals the target timbre: it plays a "Reference voice"—the TTS prompt or the VC target—next to the anonymized "Recording" and asks "How similar does the speaker in the recording sound to the reference voice?" on a 1 (Very different) to 5 (Very similar) scale, instructing raters to "judge only the speaker's voice identity, timbre, and accent" and to "ignore wording, audio quality, and loudness." The VC-only *source-preservation* task (Fig. 5c) frames the source as an "Original" and the conversion as an "Imitation" and asks, on two 1–5 scales, how well the imitation keeps the original's "timing and rhythm (pauses, speaking rate)" and its "expression and emotion," explicitly telling raters not to judge the words or whether the two voices belong to the same person.

Why quality is judged without the reference. Naturalness and quality are properties of the generated utterance alone, so we deliberately withhold the reference clip from the quality task. We expose the target timbre only in the similarity task and the source only in the preservation task, each with instructions that name the single attribute to attend to.

Deployed model and unified collection. The deployed CaDiS streaming student is the 1000-step DMD model. For every task–language–benchmark evaluation, this model and all baselines in the corresponding table were available before data collection and were rated together in one unified collection using the same protocol, 200-clip gender-balanced set, and randomized rating pages.

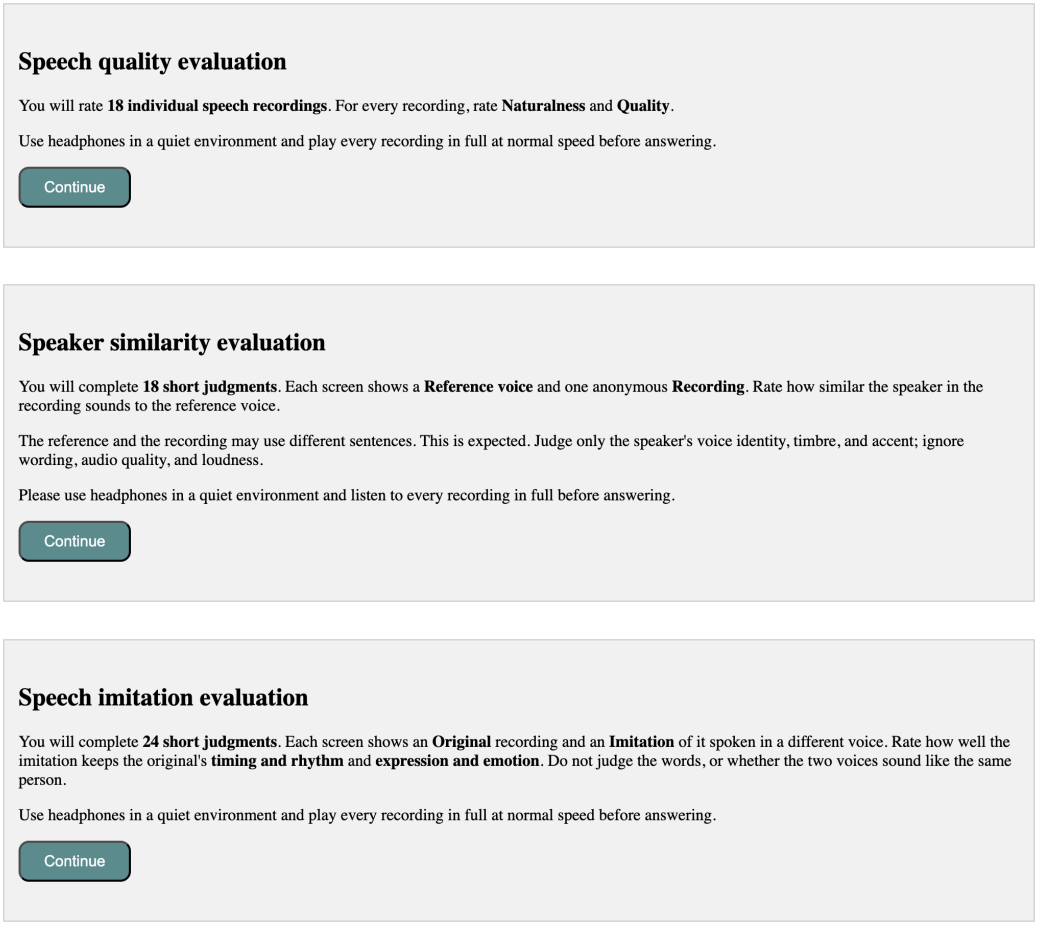
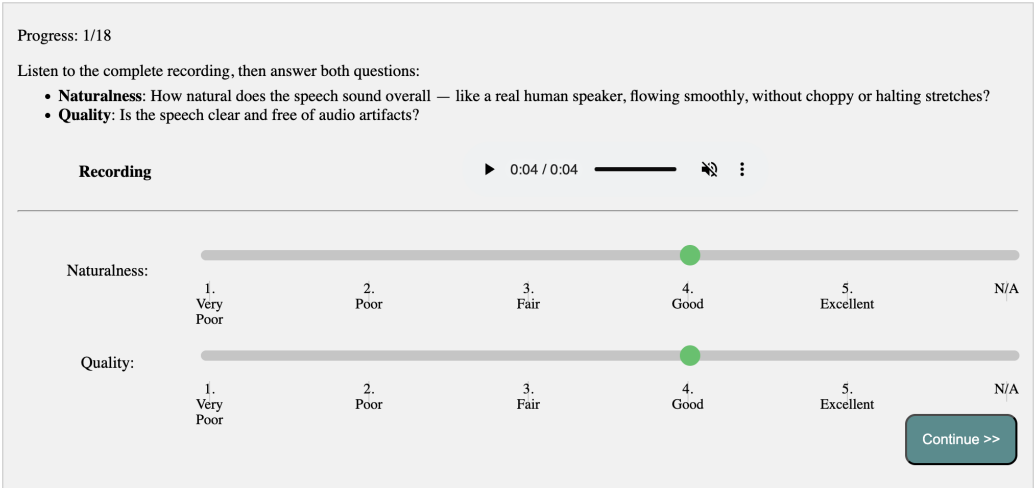
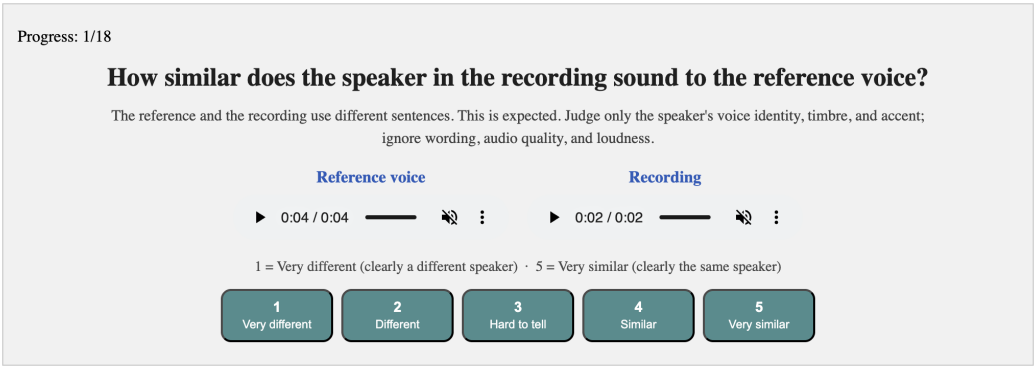


Figure 3: **Participant instructions for the English listening tasks** (welcome screens). *Top:* the naturalness/quality task rates each recording on its own. *Middle:* the similarity task shows a reference voice beside the recording and asks the rater to attend only to voice identity, timbre, and accent. *Bottom:* the VC-only source-preservation ("imitation") task asks only about timing/rhythm and expression/emotion, not the words or the speaker. TTS uses the top two instruments (18 recordings per assignment); VC uses all three (24 per assignment).

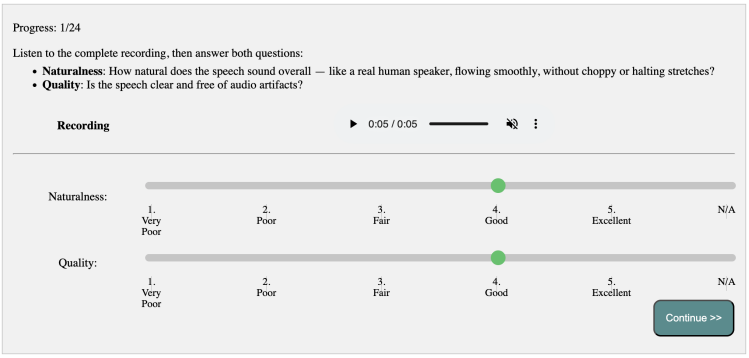


(a) Naturalness & quality (NMOS/QMOS): the recording is rated on its own, with no reference voice.

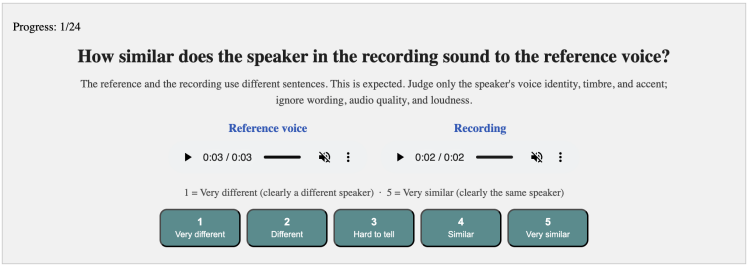


(b) Target-speaker similarity (SMOS): the prompt is shown as the "Reference voice" beside the recording.

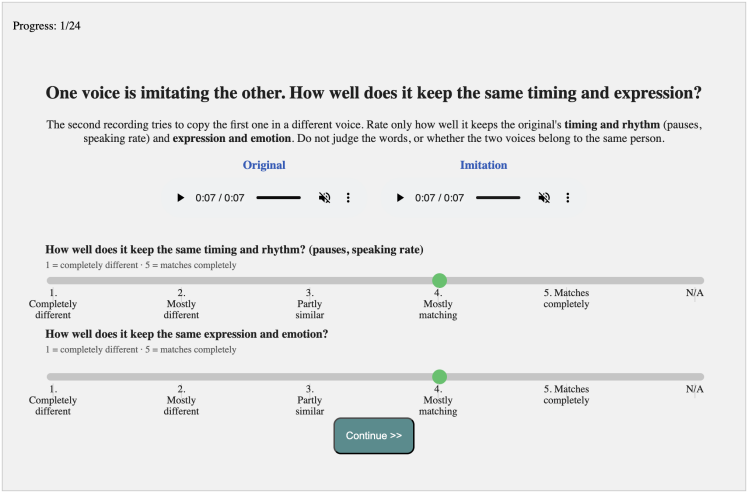
Figure 4: **English TTS listening-study interface: two tasks.** The prompt timbre appears only in the similarity task (b), never in the quality task (a), so naturalness/quality scores cannot be contaminated by the reference clip.



(a) Naturalness & quality (NMOS/QMOS): the converted output is rated on its own, no reference.



(b) Target-speaker similarity (SMOS): the conversion target is the "Reference voice."



(c) Source preservation (RHY/EXP): the source is the "Original" and the output the "Imitation."

Figure 5: **English VC listening-study interface: three tasks.** VC reuses the quality (a) and similarity (b) instruments from TTS—now 24 items per assignment and with the *target* as the reference voice—and adds a source-preservation task (c) that compares the output against the *source*. As in TTS, no reference is shown in the quality task.

Table 20: **VC subjective listening study, Anim400K English**. Columns as in Table 2. Streaming−offline (Ours): $\Delta_{\text{NMOS}}=-0.16$, $\Delta_{\text{QMOS}}=-0.09$, $\Delta_{\text{SMOS}}=-0.17$, $\Delta_{\text{RHY}}=+0.12$, $\Delta_{\text{EXP}}=+0.16$. Objective metrics in Table 3.

System	Stream	NMOS ↑	QMOS ↑	SMOS ↑	RHY ↑	EXP ↑
Offline references						
Vevo	✗	**3.83**	**3.87**	**3.76**	4.07	3.74
Ours — offline	✗	3.74	3.82	3.29	**4.13**	**3.87**
Streaming systems						
CaDiS (4-step)	✓	3.58	**3.73**	3.12	**4.25**	**4.03**
X-VC	✓	**3.60**	3.19	**3.28**	4.01	3.70
Seed-VC	✓	3.19	3.64	2.92	3.48	2.97
MeanVC-2	✓	3.28	3.20	2.59	3.78	3.36

Table 21: **TTS subjective listening study, Seed-TTS-eval Mandarin.** Columns as in Table 1; NMOS/QMOS/SMOS = naturalness/quality/prompt-similarity. Bold = best within block. The CaDiS streaming row is the deployed DMD student. Streaming−offline (Ours): $\Delta_{\text{NMOS}}=-0.21$, $\Delta_{\text{QMOS}}=-0.31$, $\Delta_{\text{SMOS}}=+0.37$. Objective metrics in Table 8.

System	Stream	NMOS ↑	QMOS ↑	SMOS ↑
Offline references				
Ours — offline	✗	**3.91**	**4.07**	3.20
F5-TTS	✗	3.51	3.82	**3.70**
IndexTTS-2.5	✗	3.47	3.65	3.61
S1-mini	✗	3.36	3.97	3.29
Streaming systems				
CaDiS (4-step)	✓	3.70	3.76	3.57
Qwen3-TTS	✓	**3.95**	**3.99**	3.40
CosyVoice3	✓	3.00	3.76	**3.78**
Faster-IndexTTS-2	✓	3.66	3.58	3.52

Confidence intervals are obtained by participant-cluster bootstrap within the same collection. Thus, every comparison is estimated directly from a common evaluation design. Table 21 likewise reports the deployed streaming student together with all Mandarin baselines from their unified collection.

For the in-domain English VC study, the streaming–offline differences are $\Delta_{\text{NMOS}}=-0.13$, $\Delta_{\text{QMOS}}=-0.14$, $\Delta_{\text{SMOS}}=+0.10$, $\Delta_{\text{RHY}}=+0.05$, and $\Delta_{\text{EXP}}=+0.01$—all small in magnitude.

Anim400K (out-of-domain) VC listening study. Table 20 reports the corresponding evaluation on anime-dub voices; the three findings from the main-text English evaluation hold here too. (i) **Streaming matches offline on every axis**: pooling both benchmarks, all ten streaming−offline differences (2 benchmarks $\times$ 5 axes) are small—for VC even the quality axis is essentially unchanged ($\Delta_{\text{QMOS}}=-0.14$ English / -0.09 anime), a stronger result than TTS. (ii) The external offline anchor **Vevo is the speaker-similarity ceiling** (SMOS 3.76 here, 3.67 English), $\sim$0.4–0.6 above our systems, while the participant-bootstrap 95% CIs for its naturalness/quality gaps to ours include zero—consistent with the DMD objective carrying no explicit speaker term. (iii) On **source preservation our streaming student leads** (best-in-block RHY/EXP on both benchmarks): the live source drives content and prosody, the flip side of its lower target similarity.

Mandarin (Seed-TTS-eval) listening study. A global native/fluent-Mandarin Prolific audience rated all systems together in a single collection; TTS results are in Table 21 (8 systems, no VoXtream) and VC in Table 22. The streaming≈offline conclusion holds throughout: the five VC axes and the TTS naturalness axis show only small differences, and only the TTS quality/artifact axis carries a clear (now-familiar) streaming cost ($\Delta_{\text{QMOS}}=-0.31$; naturalness $\Delta_{\text{NMOS}}=-0.21$), while on target similarity the deployed streaming student in fact *exceeds* the offline model ($\Delta_{\text{SMOS}}=+0.37$;

Table 22: **VC subjective listening study, Seed-TTS-eval Mandarin** (subjective MOS, n=200 gender-balanced subset). Columns as in Table 2; SMOS is vs. the target speaker, RHY/EXP are source rhythm/expression preservation. Bold = best within block. Streaming−offline (Ours): $\Delta_{\text{NMOS}}=-0.12$, $\Delta_{\text{QMOS}}=-0.07$, $\Delta_{\text{SMOS}}=-0.12$, $\Delta_{\text{RHY}}=+0.06$, $\Delta_{\text{EXP}}=+0.22$. Objective metrics in Table 9.

System	Stream	NMOS ↑	QMOS ↑	SMOS ↑	RHY ↑	EXP ↑
Offline references						
Vevo	✗	**3.44**	**3.78**	**3.87**	**3.90**	**3.66**
Ours — offline	✗	3.43	3.76	2.97	3.89	3.57
Streaming systems						
CaDiS (4-step)	✓	3.31	**3.69**	2.86	**3.95**	**3.79**
Seed-VC	✓	3.19	3.30	2.99	3.66	3.42
X-VC	✓	3.17	2.63	**3.11**	3.79	3.37
MeanVC-2	✓	**3.39**	3.43	2.69	3.78	3.52

see the prompt-similarity note below). On VC, even the quality axis is essentially unchanged ($\Delta_{\text{QMOS}}=-0.07$) and our streaming student again leads source preservation (best-in-block RHY 3.95 and EXP 3.79), while Vevo remains the speaker-similarity ceiling (SMOS 3.87). The one Mandarin-specific effect is a TTS prompt-similarity gap on the *offline* system, which sits last on SMOS (3.20) despite competitive objective WavLM-SV.

K IMPLEMENTATION AND REPRODUCIBILITY DETAILS

Few-step sampler and noise grid. At inference the four-step students denoise on a uniform-in-t grid with endpoints $t \in \{1, \frac{3}{4}, \frac{1}{2}, \frac{1}{4}, 0\}$, snapped onto the offline teacher's 32-step DDIM grid ($\{32, 24, 16, 8, 0\}/32$). Because t indexes the variance-preserving shifted-cosine schedule (shift 0.5), these equally spaced t values are *not* equally spaced in noise level; the schedule places them denser toward high SNR. The student update is stochastic rather than deterministic (DDIM): at each endpoint $t_i \to t_{i-1}$ the student predicts the clean latent $F_\theta(\mathbf{z}_{t_i}, t_i)$ and *re-noises* it to t_{i-1} with freshly sampled Gaussian noise, with the terminal step $t = 0$ returning the clean prediction directly. The Stage-3 DMD self-rollout uses this same grid and update rule, so training and deployment share one sampler.

Distillation chain. The chain is a verified init-from sequence: offline bidirectional DiT → 32-step AR causal teacher (Stage 1) → 4-step consistency student (Stage 2) → 4-step DMD student (Stage 3). For DMD, the real (frozen) score is the offline teacher of the corresponding task (the bidirectional generator for TTS, the offline content-addition converter for VC), and the DMD gradient is applied directly to the selected generated chunks, while cached history is pinned as detached conditioning. The consistency student initializes the DMD student (asymmetric initialization).

Sequence lengths and self-rollout. Target lengths are capped per task and stage (at 40 Hz, 40 frames = 1 s). For *TTS*, the AR teacher and the consistency student are trained on targets up to 30 s (1200 frames), while the Stage-3 DMD student is distilled with self-rollout on 15 s clips (600 frames); every valid chunk receives a direct DMD gradient. For *VC*, the teacher and the consistency student are trained on targets up to 15 s (600 frames), and the DMD student likewise rolls out over 15 s, but only the 24 most recent 400 ms blocks (∼9.6 s) receive direct DMD gradients. In both tasks, each generated chunk is detached before being committed to the causal cache, so these direct gradients do not propagate through earlier chunks as recurrent history. Thus, "all chunks" for TTS and "the last 24 chunks" for VC describe the support of the loss, not cross-chunk back-propagation.

Training recipe. The offline bidirectional teacher (DiT backbone, 0.8B parameters) is trained to 500k steps. From it we train the 32-step AR causal teacher (Stage 1) for 50k steps on 32 GPUs, distill the 4-step consistency student (Stage 2) for 30k steps, and reach the 4-step DMD student (Stage 3), initialized from the consistency student, in only 1000 steps; this 1000-step Stage-3 budget is the same for both the TTS and VC students. All three causal stages share the 400 ms block

configuration above; Stages 2 and 3 use four NFEs, while Stage 1 uses 32. Table 23 gives the task- and stage-specific optimization settings. We use AdamW with weight decay 0.01 throughout and the default $(\beta_1, \beta_2)=(0.9, 0.999)$, except for the DMD critic, which uses $(\beta_1, \beta_2)=(0, 0.999)$. For TTS, each microbatch concatenates examples from four data sources; the parenthesized values show their contributions. At Stage 3, four single-example microbatches are accumulated into one local batch. The reported global batch is the number of examples contributing to one optimizer update after accumulation.

Table 23: Training optimization settings. For DMD, the two learning rates are for the generator and critic, respectively. "Accum." is the number of microbatches per optimizer update.

Task	Stage	Learning rate	Microbatch/GPU	Accum.	GPUs	Global batch
TTS	Offline teacher	1×10^{-4}	36 (8+8+12+8)	1	–	–
TTS	AR teacher	3×10^{-5}	18 (4+4+6+4)	2	32	1152
TTS	Consistency	2×10^{-6}	18 (4+4+6+4)	2	32	1152
TTS	DMD	2×10^{-6} / 4×10^{-7}	1	4	64	256
VC	Offline teacher	1×10^{-4}	8	1	–	–
VC	AR teacher	1×10^{-4}	8	1	32	256
VC	Consistency	4×10^{-6}	4	1	32	128
VC	DMD	2×10^{-6} / 4×10^{-7}	1	1	64	64

Latency measurement. All FPL/RTF numbers are measured on an idle A100-80GB, with warm-up iterations discarded and the median reported ($n\geq44$). We report FPL as algorithmic buffering (chunk + each system's own look-ahead) plus measured compute-to-first-packet; RTF is wall time divided by audio duration. CPU latency numbers from prior work are not directly comparable to these GPU numbers, and we always name the hardware.

Scoring. All WER/CER/SIM numbers flow through the Seed-TTS-eval scripts (Whisper-large-v3; Paraformer-zh; WavLM-large-SV) plus Resemblyzer, SimAM-ResNet100/VoxBlink2, and UTMOS. Long-form drift uses an overlap-window ASR to avoid a non-overlapping-chunk artifact, and windowed Resemblyzer for the early→late similarity slope.