MEANFLOW FOR ONE-STEP STREAMING TEXT-TO-SPEECH

Anonymous authors
Paper under double-blind review

ABSTRACT

A text-to-speech service must start speaking soon after each request arrives while batching many requests on each GPU. Non-autoregressive flow-matching models denoise the whole utterance at every step, so they stream nothing until they finish and their latency grows with the batch; autoregressive models stream, but generating token by token takes many sequential steps or cascades of models. We study a simpler design: a single causal model that generates speech as blocks of continuous codec latents, each in one to three network evaluations, trained from scratch with a MeanFlow objective. We find that the block size trades generation difficulty for speed, that classifier-free guidance must be built into the training target and pushes the output away from real speech, and that a short fine-tuning stage with a critic in a pretrained speech representation improves one-step speaker similarity and quality at every block size at no inference cost. Trained on LibriSpeech, the fine-tuned model achieves near-human intelligibility in one step, produces its first audio in under 20 ms, and delivers more than four times the peak throughput of recent flow-matching models at their fastest intelligible settings.

1 INTRODUCTION

A text-to-speech service has to start speaking soon after each request arrives, and at high request volume it has to batch many requests on each GPU to raise its throughput. The two dominant model families handle this trade-off differently. Non-autoregressive flow-matching models (Le et al., 2024; Eskimez et al., 2024; Chen et al., 2025) generate the whole utterance over tens of network function evaluations (NFE). They produce high-quality speech, but no audio is available until the last evaluation finishes, and because every evaluation processes the full utterance, their latency grows with the batch. Autoregressive models stream, but generating speech token by token takes many sequential steps, and practical systems reduce this cost with cascades of semantic and acoustic models (Guo et al., 2025a; Du et al., 2024) or stacked transformers over interleaved token streams (Torgashov et al., 2026), which work but add substantial complexity.

We study a simpler design (fig. 1c): a single causal model that generates speech block by block as continuous codec latents, conditioned on the transcript and on the audio generated so far. It streams by construction and avoids the token rate, but with flow matching each block would need many NFE. We therefore train it with MeanFlow (Geng et al., 2025a;b), which learns average velocities so that a

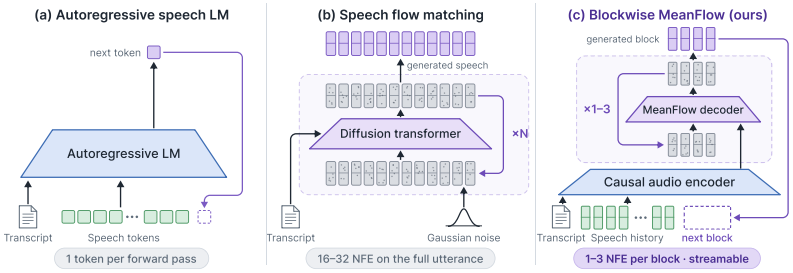

Figure 1: **Generation paradigms.** (a) An autoregressive speech language model, one token per forward pass. (b) Flow matching over the whole utterance in 32 to 64 NFE, with no streaming. (c) Blockwise MeanFlow (ours): one to three NFE per block, streaming at block rate.

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.

block can be generated in one to three NFE, from scratch and without a teacher. Few-step objectives have been established mainly for class-conditional images (Song et al., 2023); speech is harder for them because the output has to render a transcript completely and in order.

We find three things that decide whether this design works. *Block size trades generation difficulty for speed* (section 5): larger blocks amortise every network pass over more audio, but their flows are more curved, and intelligibility, speaker similarity and quality fall with the block; at every block size, few-step MeanFlow matches or beats many-step flow matching. *Guidance must be built into the training target* (section 6): classifier-free guidance is needed for intelligibility, and we give the network the guidance scale as an input, as improved MeanFlow does, but build the target from both branches at zero guidance, which is simpler and trains a better one-step model than its recursive target. Beyond a moderate scale, guidance pushes the output away from real speech, raising its level and distorting its spectrum. *A critic repairs the one-step destination* (section 7): a short fine-tuning stage with a critic in a pretrained speech representation compares generated blocks with real speech, which the regression objectives never do, and improves one-step intelligibility, speaker similarity and quality at every block size against a matched control.

Trained on the 960 hours of LibriSpeech, the block-8 model with the critic matches the intelligibility of the recordings resynthesized through its codec at one NFE per block, reaches its first audio in under 20 ms, and produces more than four times the peak throughput of F5-TTS at its fastest intelligible setting (section 8). Its speaker similarity remains below that of open systems trained on one to two orders of magnitude more speech, which is its main limitation.

2 Related Work

Few-step generation and MeanFlow in speech. Consistency models (Song et al., 2023), shortcut models (Frans et al., 2025) and MeanFlow (Geng et al., 2025a;b) train few-step generators from scratch, and distillation learns them from a teacher (Salimans & Ho, 2022; Yin et al., 2023); all are established mainly for class-conditional images. In text-to-speech the few-step objective also has to carry the alignment to a transcript. SplitMeanFlow (Guo et al., 2025b) trains a JVP-free interval-splitting consistency, starting from a pretrained flow-matching teacher, on the acoustic stage of a cascade whose semantic tokens are already aligned to the text, and IntMeanFlow (Wang et al., 2026) distils from a flow-matching teacher. We train the MeanFlow identity with its JVP from scratch on the text-conditioned problem, with text guidance built into the objective.

Autoregressive speech generation. Codec language models generate discrete tokens frame by frame, using cascades or stacked transformers to handle the token rate (Wang et al., 2023; Guo et al., 2025a; Torgashov et al., 2026), while full-utterance flow-matching models (Le et al., 2024; Eskimez et al., 2024; Chen et al., 2025) give up streaming. CALM and Pocket TTS (Rouard et al., 2026) generate one continuous Mimi frame per step with a small consistency head. DiTAR (Jia et al., 2025), the closest design to ours, conditions a diffusion transformer on a causal language model to generate patches of four continuous latent frames with ten NFE per patch. StreamMel (Wang et al., 2025) follows a related design, and concurrent CuteTTS (Zhang et al., 2026) distils guidance and sampling steps into a patch-level student. We take on the harder blockwise setting: the MeanFlow decoder generates a whole block jointly in one to three NFE, from 0.32 s to 2.56 s of speech across our sweep and 0.64 s at our default block size of 8.

Guidance and adversarial supervision. Classifier-free guidance (Ho & Salimans, 2022) oversaturates in image diffusion, and a line of work mitigates this (Saharia et al., 2022; Kynkäänniemi et al., 2024; Chung et al., 2025); in speech, strong guidance lowers speaker similarity at few sampling steps (Le et al., 2024). Guidance scales have been folded into distilled image models as an input (Meng et al., 2023); as in iMF, the scale here is part of the from-scratch objective. Vocoders are trained adversarially (Kong et al., 2020), StyleTTS 2 (Li et al., 2023) uses a discriminator on pretrained speech features, and adversarial terms appear in diffusion distillation (Yin et al., 2024). Our critic judges the one-step destinations of a trained model, spliced into real context and decoded, in a frozen self-supervised representation.

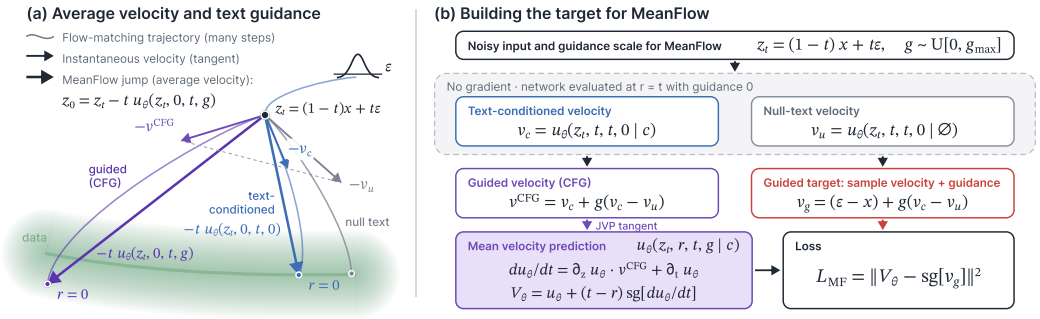

Figure 2: **Average velocity and the guided target.** (a) Instantaneous velocities at z_t (short arrows), the trajectories they generate (thin lines) and one-step MeanFlow jumps (thick arrows). (b) The guided target for one training sample; the dashed box holds the two evaluations without gradient.

3 METHOD

Blockwise MeanFlow (fig. 1c) generates speech as continuous codec latents, block by block: a causal audio encoder reads the transcript and the speech history, and a MeanFlow decoder turns Gaussian noise into the next block in one to three NFE. We first describe the few-step objective, MeanFlow (Geng et al., 2025a) in its improved form (Geng et al., 2025b), and how we build text guidance into it (fig. 2). We then describe the network (fig. 3) and the critic fine-tuning that follows training (fig. 4).

Flow matching: many small steps along a curved path. Flow matching (Lipman et al., 2023; Albergo & Vanden-Eijnden, 2023; Liu et al., 2023) moves Gaussian noise ϵ (top right of fig. 2a) to the data (green band) along a time t that runs from $t = 1$ at the noise to $t = 0$ at the data. Training places a point on the straight line between a data point x and noise, $z_t = (1-t)\,x + t\,\epsilon$, and regresses a network onto the line's velocity $\epsilon - x$. The network learns the average over all pairs whose line passes through z_t, the *instantaneous velocity*

$$v(z_t, t) = \mathbb{E}\big[\epsilon - x \mid z_t\big]. \tag{1}$$

The pairs consistent with z_t change as it moves, so the trajectory that integrates v from $t = 1$ to $t = 0$ is curved. It is the thin blue line in fig. 2a, whose tangent at z_t is the short arrow $-v_c$; arrows point along $-v$ because generation runs backwards in t. A large step along the tangent leaves the curve, so flow matching needs many small steps.

MeanFlow: one jump along the average velocity. MeanFlow (Geng et al., 2025a) instead predicts the straight jump from z_t to the point z_r that the trajectory reaches at an earlier time r. With $r = 0$ this is the thick blue arrow, which lands in one step where the blue trajectory ends. The jump divided by its duration is the *average velocity*, the mean of v along the curve:

$$u(z_t, r, t) \triangleq \frac{1}{t-r}\int_r^t v(z_\tau, \tau)\,\mathrm{d}\tau, \qquad z_r = z_t - (t-r)\,u(z_t, r, t). \tag{2}$$

As r approaches t the jump collapses onto the tangent, $u(z_t, t, t) = v(z_t, t)$. One network conditioned on r is therefore a flow-matching model at $r = t$, a one-step generator at $r = 0$, and a few-step sampler in between.

The MeanFlow identity: a target without integrating. The target for u is an integral along the very curve that takes many steps to trace. MeanFlow replaces it with a condition at a single point. Keep z_r fixed and slide the start point back along the trajectory: the jump $(t-r)\,u$ grows by exactly the distance the start point moved, $v\,\mathrm{d}t$. Hence $\frac{\mathrm{d}}{\mathrm{d}t}\big[(t-r)\,u\big] = v$, which rearranges to the *MeanFlow identity*:

$$u(z_t, r, t) = v(z_t, t) - (t-r)\,\frac{\mathrm{d}}{\mathrm{d}t}u(z_t, r, t), \qquad \frac{\mathrm{d}}{\mathrm{d}t}u = v\cdot\partial_z u + \partial_t u. \tag{3}$$

The average velocity u is the instantaneous velocity v minus a correction for how the average changes as the start point moves. Both terms come from the network: v is its output at $r = t$, and $\mathrm{d}u/\mathrm{d}t$ is the derivative of its output along $(v, 1)$, which one Jacobian-vector product (JVP) computes.

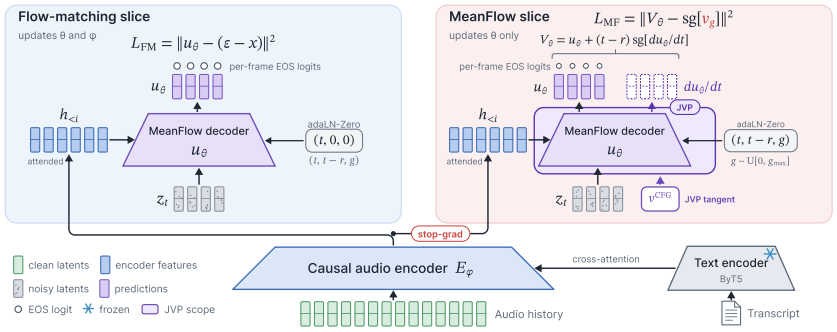

Figure 3: **Architecture and one training step.** The causal audio encoder E_φ (bottom) feeds the MeanFlow decoder u_θ (top). Left: the flow-matching slice ($r = t$, $g = 0$) updates θ and φ. Right: the MeanFlow slice takes one JVP with tangent v^{CFG} and updates θ only.

Improved MeanFlow: a regression target. Improved MeanFlow (Geng et al., 2025b) moves the correction to the network's side, forming the compound prediction

$$V_\theta = u_\theta + (t - r)\ \mathrm{sg}[\mathrm{d}u_\theta/\mathrm{d}t]\,, \tag{4}$$

the network's jump converted back into a velocity at z_t, which by eq. (3) should equal v. It regresses V_θ onto $\epsilon - x$, the per-sample stand-in for v that flow matching also uses (eq. (1)). The JVP's tangent is the network's own instantaneous prediction $u_\theta(z_t, t, t)$, and the derivative is held under stop-gradient. At $r = t$ the correction vanishes and the loss is flow matching, so one network trains both objectives. We train it from scratch, without a teacher; what we change is how guidance enters.

Text guidance changes the trajectory. Text-to-speech models almost universally apply classifier-free guidance on the text (Le et al., 2024; Lee et al., 2024; Eskimez et al., 2024; Chen et al., 2025) to ensure transcript alignment. At z_t in fig. 2a, the text-conditioned velocity v_c (blue) and the null-text velocity v_u (gray) differ, and guidance extrapolates past v_c along their difference (dashed) to the guided velocity v^{CFG} (purple). Over many steps the three fields trace three curved trajectories (thin lines) that end at different points. A one-step jump (thick arrows) must land where its trajectory ends, and that endpoint depends on the whole bent path, not only on its first direction. A one-step model trained without guidance therefore cannot be guided after the fact. We instead give the MeanFlow decoder a guidance input g and train the identity on the guided field.

Building the target for MeanFlow. Figure 2b builds the target for one training sample with interval (r, t). *Noisy input and guidance scale:* the sample has $z_t = (1 - t)\,x + t\,\epsilon$ and $g \sim \mathcal{U}[0, g_{\max}]$. *Text-conditioned and null-text velocities:* two evaluations of the MeanFlow decoder at $r = t$ and guidance 0, without gradient (dashed gray box), give

$$v_c = u_\theta(z_t, t, t, 0 \mid c), \qquad v_u = u_\theta(z_t, t, t, 0 \mid \varnothing). \tag{5}$$

Guided velocity and guided target: the guidance direction $v_c - v_u$, scaled by g, is added to v_c for the guided velocity (purple box) and to the sample velocity $\epsilon - x$ for the guided target (red box):

$$v^{\mathrm{CFG}} = v_c + g\,(v_c - v_u), \qquad v_g = (\epsilon - x) + g\,(v_c - v_u). \tag{6}$$

The guided target substitutes $\epsilon - x$ for its expectation v_c, as improved MeanFlow does for v. *Mean velocity prediction:* the decoder evaluates $u_\theta(z_t, r, t, g \mid c)$, and its JVP takes v^{CFG} as the tangent, so the identity is differentiated along the guided field; together they form V_θ (shaded box). *Loss:* the compound prediction is regressed onto the guided target,

$$\mathcal{L}_{\mathrm{MF}} = \left\| V_\theta - \mathrm{sg}[v_g] \right\|^2. \tag{7}$$

At inference, one NFE at the chosen g returns the guided average velocity. Improved MeanFlow instead builds its target from the network's own guided branch (its recursive target); taking both branches at $g = 0$ is simpler and trains a better one-step model (sections B and 6).

Causal audio encoder and MeanFlow decoder. MeanFlow needs a JVP through the network at every training step, which roughly doubles the cost of a flow-matching step. Autoregressive generation makes most of it avoidable: when block i is denoised, everything before it is clean. We therefore split the model in two (fig. 3). The *causal audio encoder* E_φ (bottom) reads the clean latents of the history (green) under causal self-attention, cross-attends to the transcript embedded by the frozen ByT5 text encoder, and produces one feature per frame (blue) without ever seeing a noise

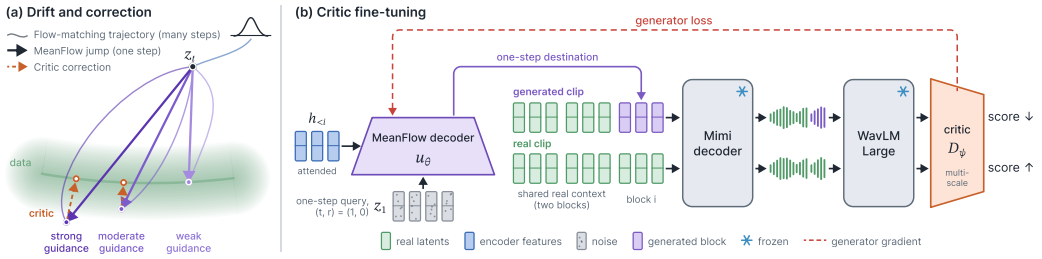

Figure 4: **Critic fine-tuning.** (a) Stronger guidance pushes one-step jumps further from the data; the critic pulls them back. (b) One fine-tuning step: the generated block, after two real context blocks, and the real clip pass through the frozen Mimi decoder and WavLM Large to the critic D_ψ.

level. The *MeanFlow decoder* u_θ (purple) takes the noisy latents z_t of block i (gray) and attends to the encoder features $h_{<i}$ of every earlier frame, with learned register tokens for the first block. It receives $(t, t-r, g)$ through adaLN-Zero and predicts the block's average velocity

$$u_\theta(z_t, r, t, g \mid c), \tag{8}$$

where c is the transcript, along with per-frame end-of-speech (EOS) logits (circles). A speaker prompt is simply the first blocks of the history, so zero-shot prompting needs no inpainting. In training, E_φ runs once over the whole clean utterance, and all blocks are denoised in parallel under a block-causal mask.

Flow-matching slice and MeanFlow slice. Each batch is split into two slices, so that one network trains on both objectives at every step. The *flow-matching slice* (fig. 3, left; seven eighths of the rows) conditions on $(t, 0, 0)$, where the loss reduces to flow matching,

$$\mathcal{L}_{\mathrm{FM}} = \left\| u_\theta(z_t, t, t, 0 \mid c) - (\epsilon - x) \right\|^2, \tag{9}$$

and updates both θ and φ. It drops the transcript with probability 0.1 to train the null-text branch that classifier-free guidance needs (Ho & Salimans, 2022), which supplies v_u in eq. (5). The *MeanFlow slice* (right) draws an interval $r < t$ and $g \sim \mathcal{U}[0, g_{\max}]$ and computes $\mathcal{L}_{\mathrm{MF}}$. Its JVP runs through the decoder alone (purple box) with v^{CFG} as the tangent, so the MeanFlow overhead is paid per block rather than over the utterance. Its encoder features pass a stop-grad (red tag), which keeps its backward pass inside the decoder. The EOS logits are trained with a per-frame binary cross-entropy, and each row's regression loss is weighted by $1/(\ell + 10^{-3})$, with ℓ its loss under stop-gradient, following MeanFlow (Geng et al., 2025a). Section A gives the interval distribution and the full architecture and training details.

Drift and correction. A network trained on finite data only approximates the true fields, so its one-step destination deviates from the data. Classifier-free guidance repairs much of this but adds distortions of its own, as it does in image diffusion (Saharia et al., 2022; Kynkäänniemi et al., 2024; Chung et al., 2025); section 6 measures both on speech. Neither slice looks at the destination: both regress onto a per-sample velocity, and the MeanFlow supervision is local, anchored at $r = t$. Adversarial training supervises the destination directly. A discriminator, the critic D_ψ, learns to tell generated samples from real ones, and the generator learns to fool it (Goodfellow et al., 2014). Adversarial training has been effective for speech generation (Kong et al., 2020; Li et al., 2023), and we use it as a short fine-tuning stage. In fig. 4a, one-step jumps from z_t (purple, darker for stronger guidance) land further from the data as guidance grows, and the critic's correction (orange dashed) points them back.

Critic fine-tuning. The critic should judge what the model produces at inference: a one-step block from pure noise, heard in context. That is the query $(t, r) = (1, 0)$, which the MeanFlow slice almost never draws, so during fine-tuning we set a quarter of its rows to it (fig. 4b). The one-step destination $z_1 - u_\theta(z_1, 0, 1, g \mid c)$ (purple) follows two real context blocks (green) in the generated clip. The real clip is the same three blocks of the recording, so only the block and its join to real speech tell the two apart. Each clip passes through the Mimi decoder in one call and is judged in the features of a self-supervised speech model, as in past work (Li et al., 2023; Chen et al., 2022): frozen WavLM Large (layer 14, star) embeds it, and a small multi-scale convolutional critic (orange) scores it per frame. Level is trivial to correct at inference, so we scale each decoded block to the RMS of its context, real and generated alike, and loudness is not a cue. The critic is trained with the

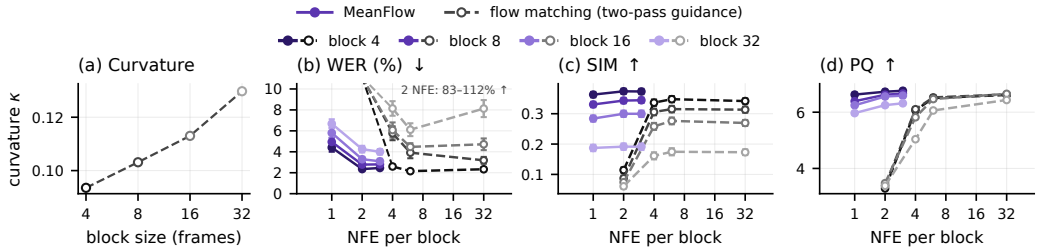

Figure 5: **Larger blocks have more curved flows and are harder at every NFE.** (a) Curvature κ of the flow-matching arm. (b–d) Quality against NFE per block. Purple: MeanFlow ($g = 2.5$); gray, dashed: flow matching with two-pass guidance ($g = 2$). Darker is a smaller block (legend).

hinge loss (Lim & Ye, 2017; Miyato et al., 2018), and the decoder with an adversarial loss $\mathcal{L}_{\text{adv}}$, the negated critic score of the generated clip (section C). For 10k steps the MeanFlow decoder minimises $\mathcal{L}_{\text{FM}} + \mathcal{L}_{\text{MF}} + \lambda\,\mathcal{L}_{\text{adv}}$ with $\lambda = 0.003$; the gradient of $\mathcal{L}_{\text{adv}}$ reaches u_θ through the frozen WavLM Large and Mimi decoder (red dashed). Fine-tuning adds nothing at inference. We generate a block from $z_1 \sim \mathcal{N}(0, I)$ in S MeanFlow steps on a uniform grid in t, each querying $u_\theta(z_t, t_{\text{next}}, t, g \mid c)$ and jumping to t_{next}; $S = 1$ is the single jump of fig. 2a. Generation stops when the EOS logit fires, and the Mimi decoder decodes each block as it lands. The time to first audio is therefore one pass of the text and audio encoders over the prompt, S NFE of the MeanFlow decoder, and one block decode.

4 Experimental Setup

All models are trained on the 960 hours of LibriSpeech (Panayotov et al., 2015). Speech is represented by the latents of the Mimi variational autoencoder (Défossez et al., 2024; Rouard et al., 2026), 32 channels at 12.5 Hz, so a block of 8 frames is 0.64 s; text is encoded by ByT5-base (Xue et al., 2022), and both stay frozen. The causal audio encoder has 8 layers of width 512 and the MeanFlow decoder 8 layers of width 1024. Unless a section varies it, the block size is 8, the flow-matching slice is seven eighths of each batch, and the guidance input is drawn up to $g_{\max} = 5$. Each model trains once, for 100k steps at batch 256 with Muon (Jordan et al., 2024) on one GPU, and we evaluate its EMA weights (section A). We evaluate on the LibriSpeech-PC test-clean cross-sentence set of the F5-TTS protocol (Chen et al., 2025; Meister et al., 2023): 1127 utterances, each generated from a transcript and a prompt recording of the same speaker from a different utterance. We report the word error rate (WER) of a Whisper large-v3 transcription (Radford et al., 2023), speaker similarity (SIM) between WavLM Large ECAPA embeddings (Chen et al., 2022) of the output and the prompt, and production quality (PQ) from audiobox-aesthetics (Tjandra et al., 2025), which does not compare against a reference, so a model can exceed the recordings on it. Unless stated, $g = 2.5$ for the MeanFlow models and $g = 2$ for the flow-matching baselines, which pay two NFE per step. Our outputs are decoded by Mimi, so their reference is the recording after a round trip through the codec (table 5). Error bars in figures are ± 1 standard error and intervals in tables are 95 %, both from a speaker-cluster bootstrap over the 39 test speakers, paired by utterance for contrasts between models.

5 Block Size Trades Generation Difficulty for Speed

The block size pulls in two directions. A large block is fast: every pass of the causal audio encoder, the MeanFlow decoder and the Mimi decoder covers more audio, so throughput grows with the block (table 6 in section D). A large block is also hard. At 32 frames a block is two and a half seconds of speech, several words with their timing and intonation, and the MeanFlow decoder has to resolve all of it at once, ideally in one NFE. A small block is a fraction of a second, much of it constrained by the history just before it. We measure what a larger block costs, first in the curvature of the per-block flow and then in quality against NFE, for both objectives (fig. 5).

The block sets the curvature of the flow. We measure how curved each block's flow is with a straightness measure from prior work (Liu et al., 2023; Zhao et al., 2025). Given a block's real history, we integrate the flow-matching arm's ODE from noise ϵ to a sample $\hat{x}$ in $K = 32$ Euler steps, without

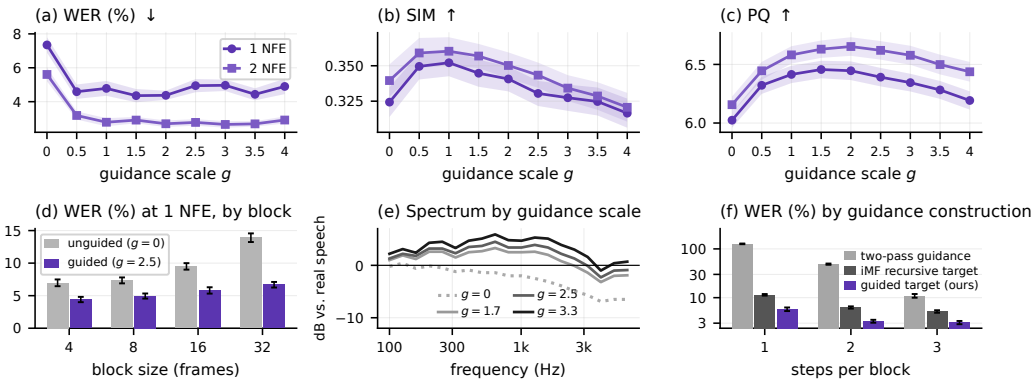

Figure 6: **Guidance scale and construction.** (a–c) The block-8 MeanFlow model at one and two NFE against g. (d) One-NFE WER with and without guidance, by block size. (e) Spectrum of the one-NFE output relative to the recordings. (f) Three guidance constructions (block 16; two-pass guidance at one step is above 100 %).

guidance, and compare each velocity along the way with that of the straight line between the two ends:

$$\kappa = \mathbb{E}_\epsilon\Big[\frac{1}{K}\sum_{k=1}^{K}\frac{1}{d}\left\|v_\theta(z_{t_k}, t_k) - (\epsilon - \hat{x})\right\|^2\Big], \qquad t_k = k/K, \tag{10}$$

where d is the number of latent dimensions in the block; a straight flow has $\kappa = 0$. The curvature rises steadily from block 4 to block 32 (fig. 5a). The picture of flow matching in section 3 suggests why. Near the noise end, the velocity averages over every clean block consistent with the noisy point, and for a long block these differ in which words come next and how they are said, so the trajectory bends as it commits to one of them. A more curved path is harder to cover in few steps.

Few-step MeanFlow matches many-step flow matching, and larger blocks are harder for both. Figure 5b–d put both objectives on one axis, NFE per block, with guidance passes counted. Flow matching produces noise-like output at one step and needs four to six NFE to be usable. MeanFlow is usable at one NFE, and at two or three it reaches or beats the 32-NFE flow-matching reference on WER from block 8 upward, with block 4 close (b). On speaker similarity, MeanFlow at one NFE is at or above flow matching at any NFE for every block size (c). Quality is where one step costs MeanFlow the most (d): at three NFE it matches the reference up to block 16, and its one-step quality falls further behind as the block grows. Across blocks, speaker similarity roughly halves from block 4 to block 32 for both objectives. A large block carries the speaker across seconds of its own generated speech. WER rises with the block for both objectives, and the one-step model loses the most, as the curvature in (a) predicts. At large blocks many-step sampling is not even the better sampler: the WER of flow matching at block 32 rises from six to 32 NFE (b). Our recipe uses block 8, where one-step MeanFlow is already usable. The next two sections narrow the gap at larger blocks: guidance built into the objective makes the few-step model intelligible (section 6), and a critic repairs the one-step destination (section 7).

6 Guidance for MeanFlow

Guidance is needed for intelligibility at every block size. Figure 6 asks how much guidance a MeanFlow model needs, what it does to the output beyond intelligibility, and how it has to enter the model for one step to work. Unguided, at the left end of fig. 6a, the block-8 model's WER is about one and a half times its guided value at one NFE and twice at two. The first half-unit of guidance recovers most of this, and WER stops improving beyond $g \approx 1$, while speaker similarity and quality rise to a peak between $g = 1$ and $g = 2$ (b, c). Guidance lowers one-step WER by a third or more at every block size, and most at block 32 (d).

Strong guidance moves the output away from real speech. Beyond that peak, more guidance lowers speaker similarity and quality while WER stays flat (b, c): for $g > 0$ the guided velocity does not transport noise to data, so its destinations need not lie where real speech lies. Panel (e)

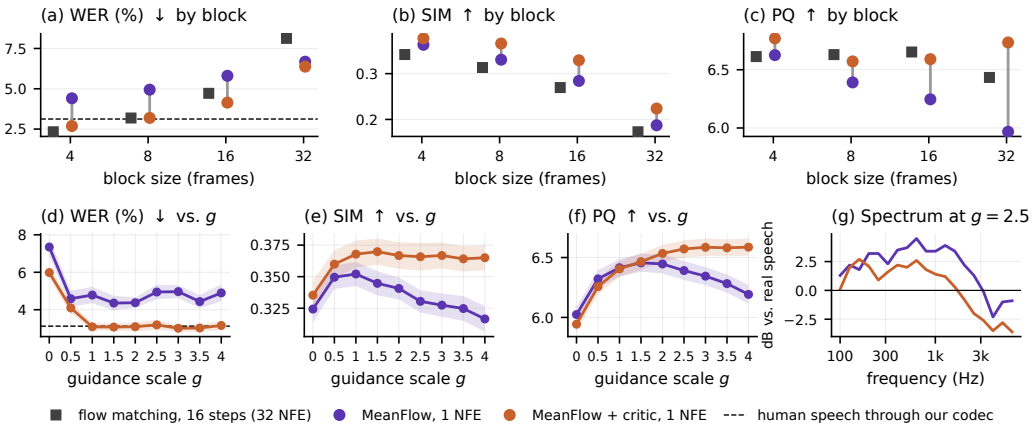

Figure 7: **One-step quality before and after critic fine-tuning.** (a–c) By block size, before (purple) and after (orange) fine-tuning; squares: flow matching at 16 steps; dashed: human speech through our codec. (d–f) Block 8 against g. (g) Spectrum relative to the recordings.

shows the drift in the spectrum relative to the recordings. Unguided (dotted), the output is quieter and falls off faster at high frequencies; each step up in guidance adds level across the speech band, up to a broad excess below about 2 kHz at $g = 2.5$ whose shape loudness normalization does not remove. This matches the over-saturation along the conditional direction described for image diffusion (Kynkäänniemi et al., 2024; Chung et al., 2025) and the speaker-similarity cost of strong guidance at few sampling steps reported for speech (Le et al., 2024); we are not aware of a prior measurement of the spectral excess in speech.

Guidance must be built into the training target. How guidance enters a MeanFlow model decides whether one step works (fig. 6f; table 3 in section B). Two-pass guidance on a model trained with the plain identity doubles the NFE of every step and still fails at one step; the iMF recursive target trains a usable model but is worse than our guided target at every step count.

7 Critic Fine-Tuning

Guidance moves the one-step output away from real speech (section 6), and the regression objectives never compare a generated block with a real one. The critic supplies that comparison. We fine-tune each block's MeanFlow model from its 100k-step weights for 10k steps with the critic, against a matched control trained the same 10k steps on the same schedule without it. After fine-tuning, the one-step model has lower WER and higher speaker similarity and quality at every block size (fig. 7a–c), with the quality gain growing to most of a point of PQ at block 32. With one NFE against 32, it is above the sixteen-step flow-matching reference on speaker similarity at every block, about level on quality, and level or better on WER from block 8 up. Against the matched control the gains hold at every block size, largest at the larger blocks (table 4). The critic also roughly halves the mid-band excess that guidance adds and brings the level within about half a decibel of the recordings (g). The critic does not undo guidance. In the guidance sweep of the block-8 model (fig. 7d–f), one-NFE WER falls with the first unit of guidance as before, and at strong guidance, where similarity and quality fall before fine-tuning, they stay flat or keep rising with the critic. The ablations (table 4) show that all three judges lower WER, and that the judge's feature space decides what else improves: a spectrogram critic raises only quality, a critic on the decoder's own features only similarity, and the WavLM Large critic both.

8 Comparison with Open Systems

We compare our block-8 model with the open systems we could run under the same protocol, two non-autoregressive flow-matching models and a streaming codec language model (fig. 8; every value and the timing protocol are in table 5 and section D). In fig. 8a each curve runs from batch 1 at its lower left to batch 256; lower and further to the right is better. Our curves, orange for the critic

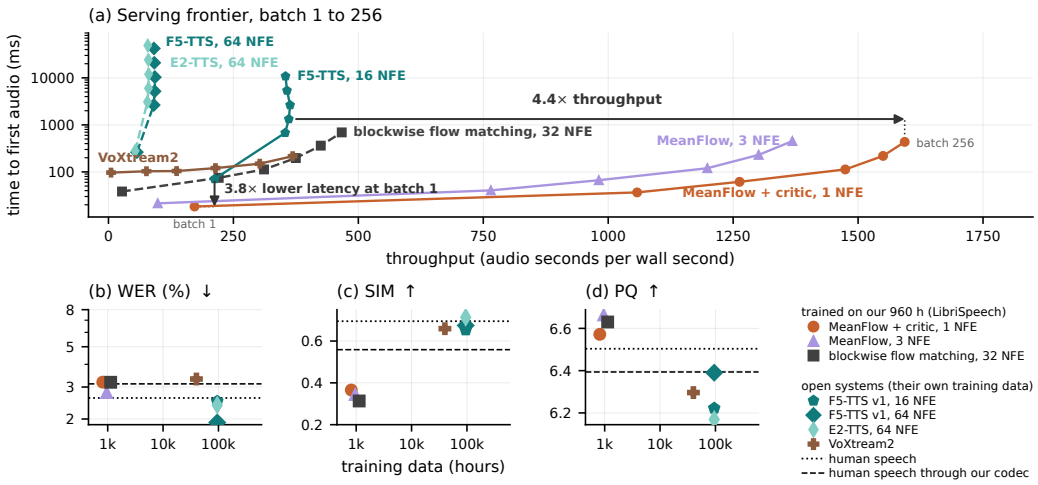

Figure 8: **Serving frontier and training data.** (a) Time to first audio against throughput on one B300, from batch 1 (lower left) to 256. Blockwise flow matching is our architecture trained with flow matching and sampled in 16 steps with two-pass guidance. Arrows: our one-step model against F5-TTS at 16 NFE. (b–d) Quality against training hours; dotted: human speech; dashed: human speech through our codec. Values: table 5.

model at one NFE and purple for the base model at three, run along the bottom of fig. 8a. Their first audio, including the text encoder, the prompt pass and the codec decode, is ready within tens of milliseconds at batch 1 and within half a second at batch 256, while their throughput grows by an order of magnitude. The non-autoregressive systems (teal) deliver audio only when the utterance is done, so they climb almost vertically: their throughput saturates by batch 16, and larger batches only lengthen every wait. Even at batch 1, F5-TTS at 16 NFE, its fewest intelligible setting (table 5), takes about four times as long as our one-step model to its first audio (vertical arrow). The streaming codec language model VoXtream2 (brown) starts about five times later at batch 1 and needs batch 256 to match the throughput of F5-TTS. At peak, our one-step model produces more than four times as much audio per second as F5-TTS at 16 NFE (horizontal arrow). Streaming alone does not buy this: blockwise flow matching (dark gray), our architecture sampled at 32 NFE, also starts early, but its throughput levels off not far above that of F5-TTS. The gain is not proportional to the NFE count, since the causal audio encoder and the codec decoder take a fixed share of every block; going from three NFE to one raises throughput by less than a factor of two.

Quality against training data. Our models are trained on 960 hours of speech, an order of magnitude or more below every open system (fig. 8b–d). At two and three NFE, and at one with the critic, their WER sits at the level of human speech through our codec (b, dashed), as VoXtream2's does, with F5-TTS and E2-TTS below it. Their quality is above that of every open system (d). Speaker similarity is where they fall short (c). The gap is not the codec, since the output of F5-TTS passed through our codec is as similar to the speaker as human speech passed through it (table 5). The open systems differ from ours in two ways that we cannot separate here: they train on far more speech, and the diffusion-based systems also guide on the speaker prompt, dropping it together with the transcript in their unconditional pass (Le et al., 2024; Eskimez et al., 2024; Chen et al., 2025). We focus on text guidance in the MeanFlow objective and leave joint guidance on the text and the speaker prompt to future work.

9 Conclusion

Blockwise MeanFlow turns few-step generation into a streaming text-to-speech system: each block of continuous codec latents comes from one to three evaluations of a decoder conditioned on a causal encoder of the text and the speech so far. It starts speaking within 20 ms, serves several times the throughput of flow-matching systems while maintaining near-human intelligiblity, and has higher production quality than recent open systems.

AI USE STATEMENT

We used large language model assistants to help write code for experiments, analysis and plotting, and to edit the text of the manuscript. The research questions, experimental design and conclusions are the authors' own, and the authors checked every result reported in the paper.

ETHICS STATEMENT

Zero-shot voice cloning from a few seconds of a speaker's recording carries the risk of impersonation. Our models are trained and evaluated on LibriSpeech, which is derived from public-domain audiobook recordings and distributed under CC BY 4.0; no new human-subject data were collected. Watermarking and synthetic-speech detection are the appropriate technical mitigations for systems of this kind.

REPRODUCIBILITY STATEMENT

We describe the architecture and training objective in section 3, and give all hyperparameters and training details in section A, the guidance constructions in section B, the critic in section C, and the timing protocol in section D. The evaluation follows the public LibriSpeech-PC protocol of F5-TTS (section 4), and all training and evaluation data are publicly available. We will release our code upon acceptance.

REFERENCES

Michael S. Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=li7qeBbCR1t.

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

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 Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pp. 6255–6271. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.acl-long.313/.

Hyungjin Chung, Jeongsol Kim, Geon Yeong Park, Hyelin Nam, and Jong Chul Ye. Cfg++: Manifold-constrained classifier free guidance for diffusion models. In *International Conference on Learning Representations*, volume 2025, pp. 30824–30850, 2025.

Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. Moshi: a speech-text foundation model for real-time dialogue. *arXiv preprint arXiv:2410.00037*, 2024.

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

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 *2024 IEEE Spoken Language Technology Workshop (SLT)*, pp. 682–689, 2024. doi: 10.1109/SLT61566.2024.10832320.

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *International Conference on Learning Representations*, volume 2025, pp. 34668–34684, 2025.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *CoRR*, abs/2505.13447, 2025a. doi: 10.48550/ARXIV.2505.13447. URL https://doi.org/10.48550/arXiv.2505.13447.

Zhengyang Geng, Yiyang Lu, Zongze Wu, Eli Shechtman, J. Zico Kolter, and Kaiming He. Improved mean flows: On the challenges of fastforward generative models. *CoRR*, abs/2512.02012, 2025b. doi: 10.48550/ARXIV.2512.02012. URL https://doi.org/10.48550/arXiv.2512.02012.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Hao-Han Guo, Yao Hu, Fei-Yu Shen, Xu Tang, Yi-Chen Wu, Feng-Long Xie, and Kun Xie. Fireredtts-1s: An upgraded streamable foundation text-to-speech system, 2025a. URL https://arxiv.org/abs/2503.20499.

Yi Guo, Wei Wang, Zhihang Yuan, Rong Cao, Kuan Chen, Zhengyang Chen, Yuanyuan Huo, Yang Zhang, Yuping Wang, Shouda Liu, et al. Splitmeanflow: Interval splitting consistency in few-step generative modeling. *arXiv preprint arXiv:2507.16884*, 2025b.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *CoRR*, 2022.

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 Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 27255–27270. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/jia25a.html.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL https://kellerjordan.github.io/posts/muon/.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in Neural Information Processing Systems*, 33:17022–17033, 2020.

Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=nAIhvNy15T.

Matthew Le, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, Rashel Moritz, Mary Williamson, Vimal Manohar, Yossi Adi, Jay Mahadeokar, et al. Voicebox: Text-guided multilingual universal speech generation at scale. 2024.

Keon Lee, Dong Won Kim, Jaehyeon Kim, and Jaewoong Cho. Ditto-tts: Efficient and scalable zero-shot text-to-speech with diffusion transformer. *CoRR*, 2024.

Yinghao Aaron Li, Cong Han, Vinay Raghavan, Gavin Mischler, and Nima Mesgarani. Styletts 2: Towards human-level text-to-speech through style diffusion and adversarial training with large speech language models. *Advances in neural information processing systems*, 36:19594–19621, 2023.

Jae Hyun Lim and Jong Chul Ye. Geometric gan. *arXiv preprint arXiv:1705.02894*, 2017.

Yaron Lipman, {Ricky T.Q.} Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. 2023. 11th International Conference on Learning Representations, ICLR 2023 ; Conference date: 01-05-2023 Through 05-05-2023.

Xingchao Liu, Chengyue Gong, and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

Aleksandr Meister, Matvei Novikov, Nikolay Karpov, Evelina Bakhturina, Vitaly Lavrukhin, and Boris Ginsburg. Librispeech-pc: Benchmark for evaluation of punctuation and capitalization capabilities of end-to-end asr models. In *2023 IEEE automatic speech recognition and understanding workshop (ASRU)*, pp. 1–7. IEEE, 2023.

Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14297–14306. IEEE, 2023.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=B1QRgziT-.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pp. 5206–5210. IEEE, 2015.

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*, 2023.

Simon Rouard, Manu Orsini, Axel Roebel, Neil Zeghidour, and Alexandre Défossez. Continuous audio language models, 2026. URL https://arxiv.org/abs/2509.06926.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *Proc. NeurIPS*, 2022.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *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 Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32211–32252. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/song23a.html.

Andros Tjandra, Yi-Chiao Wu, Baishan Guo, John Hoffman, Brian Ellis, Apoorv Vyas, Bowen Shi, Sanyuan Chen, Matt Le, Nick Zacharov, Carleigh Wood, Ann Lee, and Wei-Ning Hsu. Meta audiobox aesthetics: Unified automatic quality assessment for speech, music, and sound. 2025. URL https://arxiv.org/abs/2502.05139.

Nikita Torgashov, Gustav Eje Henter, and Gabriel Skantze. Voxtream2: Full-stream TTS with dynamic speaking rate control. *CoRR*, abs/2603.13518, 2026. doi: 10.48550/ARXIV.2603.13518. URL https://doi.org/10.48550/arXiv.2603.13518.

Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, et al. Neural codec language models are zero-shot text to speech synthesizers. *CoRR*, 2023.

Hui Wang, Yifan Yang, Shujie Liu, Jinyu Li, Lingwei Meng, Yanqing Liu, Jiaming Zhou, Haoqin Sun, Yan Lu, and Yong Qin. Streammel: Real-time zero-shot text-to-speech via interleaved continuous autoregressive modeling. *IEEE Signal Processing Letters*, 2025.

Wei Wang, Rong Cao, Yi Guo, Zhengyang Chen, Kuan Chen, and Yuanyuan Huo. Int-meanflow: Few-step speech generation with integral velocity distillation. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 18317–18321. IEEE, 2026.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. Byt5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306, 2022.

Tianwei Yin, Michael Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6613–6623, 2023. URL https://api.semanticscholar.org/CorpusID:265506768.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T Freeman. Improved distribution matching distillation for fast image synthesis. *Advances in neural information processing systems*, 37:47455–47487, 2024.

Yuqian Zhang, Yao Shi, Kexin Huang, Botian Jiang, Zhe Xu, Yiwei Zhao, Min Liang, Shuang Chen, and Xipeng Qiu. Cutetts: Efficient and high-quality speech synthesis via autoregressive modeling of continuous latents. *arXiv preprint arXiv:2608.08638*, 2026.

Qinyu Zhao, Jaskirat Singh, Ming Xu, Akshay Asthana, Stephen Gould, and Liang Zheng. Disa: Diffusion step annealing in autoregressive image generation, 2025. URL https://arxiv.org/abs/2505.20297.

A Implementation Details

Hyperparameters. Table 1 lists the base recipe. The block-size sweep changes only the block size (and the register count, which follows it); the flow-matching arms set the MeanFlow partition to zero and remove the guidance input; the SplitMeanFlow arms replace the JVP with the interval-splitting target; the non-autoregressive arms replace the encoder-decoder by a single 8-layer transformer of width 1024 over the whole latent sequence with a learned inpainting flag, keep a random clean prefix with probability 0.5 during training, and prompt with a 3 s prefix at inference.

Table 1: Base recipe.

codec	Mimi VAE (pocket-tts weights), 32 channels at 12.5 Hz, 24 kHz audio
text encoder	ByT5-base, frozen, 256 byte tokens, cross-attention in the causal audio encoder
causal audio encoder	8 layers, width 512, causal self-attention, RoPE, dropout 0.1
MeanFlow decoder	8 layers, width 1024, dense skip connections, adaLN-Zero on $(t, t-r, g)$, dropout 0.1
parameters	85M in the causal audio encoder and 246M in the MeanFlow decoder (the frozen text encoder and codec excluded)
context	full history: block i attends to the encoder features of registers and all frames before it
registers	one learnable token per block frame (8)
block size	8 frames (0.64 s) unless swept
latent normalisation	per-channel mean, one scalar standard deviation
partition	87.5 % flow-matching rows, 12.5 % MeanFlow rows, deterministic per batch
noise level	$\gamma_t \sim \mathcal{N}(-2, 3^2)$ in logSNR
jump	$\Delta\gamma \sim \mathrm{Exp}(0.3)$ in logSNR (mean 3.3)
guidance	$g \sim \mathcal{U}[0, 5]$ on MeanFlow rows; text dropout 0.1 on flow-matching rows
loss	squared error on the compound prediction, per-row adaptive weight $1/(\ell + 10^{-3})$ with ℓ the row's detached loss, on both slices
EOS	auxiliary end-of-speech head, weight 10^{-3}
optimiser	Muon, learning rate 2×10^{-4}, weight decay 10^{-3}, bf16
batch and steps	256 utterances of 4–20 s, 100k steps (about 90 epochs)
EMA	decay 0.99
JVP	compiled forward-mode JVP through the MeanFlow decoder, math-SDPA attention inside the JVP
history noising	off
representation alignment	off (re-ablated at 100k: no resolvable gain)

Critic fine-tuning. From the 100k-step weights: 10k further steps, same optimiser and batch, generator learning rate 10^{-4}, critic learning rate 10^{-4}, adversarial weight 0.003, 25 % of MeanFlow rows at $(t, r) = (1, 0)$, critic on WavLM Large layer 14 of the decoded three-block clips with

block-to-context RMS normalisation, separate gradient clipping for generator and critic. The matched control runs the same 10k steps without the critic.

Hardware and wall-clock. Each training arm runs on one B300 for 100k steps, which takes fifteen to eighteen hours at batch 256; the critic fine-tunes take about two hours. Protocol evaluation of one cell (1127 utterances, generation and scoring) takes fifteen minutes to an hour on one GPU, depending on the step count and on contention.

The interval distribution. MeanFlow training draws a pair $r \leq t$. Drawing both from the flow-matching time distribution ties the jump length to that schedule, so we draw the jump in logSNR instead: with γ_t the row's noise level, $\gamma_r = \gamma_t + \Delta\gamma$ with $\Delta\gamma \sim \mathrm{Exp}(\lambda)$, so that $r \leq t$ holds by construction and the mean jump is $1/\lambda$ in logSNR regardless of how t is scheduled; the recipe uses $\lambda = 0.3$ with $\gamma_t \sim \mathcal{N}(-2, 3^2)$. One-step generation queries the full interval, and the training distribution has to put mass on jumps of that size. Table 2 compares the recipe's exponential jump with a uniform and a deterministic jump at the same mean, the order-statistics draw of the original MeanFlow, and the recipe with a quarter of the MeanFlow rows anchored to the exact one-step pair, all trained from scratch on the block-8 model.

Table 2: The (t, r) interval distribution on the block-8 model, LibriSpeech-PC, $g = 2.5$ (corpus WER in %, SIM, PQ).

	1 step			2 steps			3 steps			16 steps		
interval distribution	WER	SIM	PQ	WER	SIM	PQ	WER	SIM	PQ	WER	SIM	PQ
exponential jump, rate 0.3 (recipe)	4.9	0.330	6.39	2.8	0.343	6.62	2.8	0.344	6.66	3.1	0.340	6.68
uniform jump, same mean	4.5	0.330	6.43	2.6	0.343	6.62	2.6	0.342	6.67	2.9	0.343	6.70
deterministic jump, same mean	17.0	0.221	3.79	2.6	0.335	6.68	2.7	0.347	6.73	2.8	0.341	6.62
order statistics (original MeanFlow)	5.1	0.325	6.43	2.9	0.345	6.66	2.8	0.347	6.71	2.8	0.349	6.75
recipe + 25 % rows at $(1, 0)$	3.4	0.332	6.38	2.5	0.347	6.62	2.5	0.348	6.68	2.8	0.350	6.72

Production quality as the quality metric. PQ is the production-quality axis of a predictor trained on human ratings, which its authors report tracks listeners better than earlier MOS predictors (Tjandra et al., 2025), and it is our quality metric. It scores production quality, not resemblance to a reference: real LibriSpeech recordings score 6.50, our models score higher through the same codec (table 5), and the gap says something about the recordings as much as about the models. We therefore read PQ as a quality axis on which models can be ordered, not as a distance to real speech. The supplementary material contains a blind listening bundle with matched utterances next to prompt and ground truth, with the per-utterance metrics shown.

B Guidance Constructions

Table 3 is the numerical record behind fig. 6f. It compares three ways of guiding a MeanFlow model of the same architecture: our guided target (section 3), whose two branches are evaluated at $g = 0$ and combined outside the network; the iMF recursive target (Geng et al., 2025b), which builds the target from the network's own guided branch; and two-pass guidance applied at inference to a model trained on the plain identity. Each is sampled with one to three MeanFlow steps and with the sixteen-step Euler sampler of the same model.

C Critic Details and Ablations

Architecture and losses. The critic sees a 1.92 s clip: the two real blocks preceding a training block and the block itself, either real or generated. Both clips are decoded through the frozen Mimi decoder in one call, resampled to 16 kHz, and embedded by a frozen WavLM Large (Chen et al., 2022); the layer-14 hidden states (the layer also used for our WavLM feature cache) feed a multi-scale one-dimensional convolutional critic with per-frame outputs. Real and generated clips share their two context blocks, so the critic's only discriminative evidence is the block and its join to the context; each block is scaled to the RMS of its own context, real and generated alike, so loudness is not a cue. Let a and $\hat{a}$ be the real and generated clips after the Mimi decoder and this scaling, and let $D_\psi^k(\cdot)$ be the critic's per-frame scores at scale k. The critic is trained with the hinge loss, which pushes real

Table 3: **Guidance has to be built into the objective.** Three ways of guiding a MeanFlow model, our guided target, two-pass guidance and the iMF recursive target (block 16, 100k steps, LibriSpeech-PC), sampled with one to three MeanFlow steps and with the sixteen-step Euler sampler of the same model; $g = 2.5$ for our guided target, $g = 1$ for the iMF recursive target (iMF's $\omega = 2$), and $g = 2$ for two-pass guidance.

Guidance construction	Sampler	WER (%) ↓	SIM ↑	PQ ↑
Guided target, 1 NFE per step (ours)				
	MeanFlow, 1 step	5.8	0.284	6.25
	MeanFlow, 2 steps	3.3	0.300	6.56
	MeanFlow, 3 steps	3.1	0.300	6.59
	Euler, 16 steps	3.9	0.295	6.66
Two-pass guidance at inference, 2 NFE per step				
	MeanFlow, 1 step	126.8	0.030	3.32
	MeanFlow, 2 steps	48.7	0.127	4.57
	MeanFlow, 3 steps	10.9	0.219	5.41
	Euler, 16 steps	5.4	0.270	6.58
iMF recursive target, 1 NFE per step				
	MeanFlow, 1 step	11.4	0.242	5.89
	MeanFlow, 2 steps	6.3	0.276	6.55
	MeanFlow, 3 steps	5.2	0.279	6.64
	Euler, 16 steps	8.8	0.281	6.69

scores up and generated scores down, and the generator with the negated score of the generated clip:

$$\mathcal{L}_D = \sum_k \mathbb{E}\big[\max(0, 1 - D^k_\psi(a)) + \max(0, 1 + D^k_\psi(\hat{a}))\big], \qquad \mathcal{L}_{\text{adv}} = -\sum_k \mathbb{E}\big[D^k_\psi(\hat{a})\big], \quad (11)$$

with expectations over clips and frames. The critic minimises $\mathcal{L}_D$; $\mathcal{L}_{\text{adv}}$ enters at weight $\lambda = 0.003$ next to the MeanFlow and flow-matching losses of the base recipe; the critic trains at learning rate 10^{-4} without warm-up, and generator and critic gradients are clipped separately. Gradients reach the MeanFlow decoder through the Mimi decoder; the codec, WavLM Large and the text encoder receive none. The fakes are the one-step destinations $z_1 - u_\theta(z_1, 0, 1)$ of exact $(t, r) = (1, 0)$ rows, which make up 25 % of the MeanFlow rows during fine-tuning (and in the matched control).

Ablations. Table 4 gives the block-8 recipe, its no-critic control and the model before fine-tuning, the recipe with one thing changed (the judge, the loss, the training rows), and the recipe at blocks 4, 16 and 32 against their own controls. Every row is one fine-tuning run of 10k steps evaluated at one NFE and $g = 2.5$.

The guidance frontier after the critic. Figure 9 shows the full frontier at one NFE for the base, the no-critic continuation and the critic fine-tune, including WER, whose three curves track each other, and the spectrum against the no-critic control.

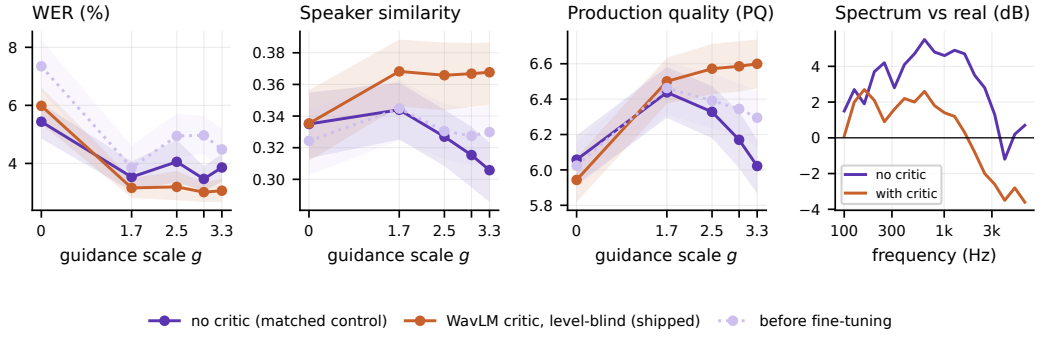

Figure 9: WER, SIM and PQ against the guidance input g at one NFE for the block-8 model before fine-tuning, after 10k steps without a critic and after 10k steps with the critic (95 % speaker-cluster intervals), and the long-term average spectrum relative to the real recordings at the operating point without and with the critic.

Table 4: **Critic fine-tuning and its ablations.** One NFE, $g = 2.5$, LibriSpeech-PC; fine-tuned rows after 10k steps from the 100k-step model. Ours uses a WavLM Large critic on clips with two real context blocks, the hinge loss, and a quarter of the MeanFlow rows at the one-step query; each control trains the same 10k steps without the critic.

	WER (%) ↓	SIM ↑	PQ ↑
Before fine-tuning	4.95	0.330	6.39
Critic fine-tuning (ours)	**3.19**	**0.366**	**6.57**
No critic, same 10k steps	4.06	0.327	6.33
Critic fine-tuning with one change			
Judge: multi-resolution STFT	3.42	0.332	6.48
Judge: MeanFlow decoder features	3.21	0.362	6.33
Loss: least squares	3.72	0.353	6.58
Training rows: one-step queries only	13.45	0.176	4.25
Critic fine-tuning at other block sizes			
Block 4, critic	2.69	0.377	6.77
no critic, same 10k steps	3.08	0.358	6.64
Block 16, critic	4.14	0.329	6.59
no critic, same 10k steps	4.77	0.275	6.12
Block 32, critic	6.37	0.224	6.73
no critic, same 10k steps	7.36	0.179	5.87

D Comparison with Open Systems: Values and Timing Protocol

Table 5 gives the values behind fig. 8: quality on LibriSpeech-PC, time to first audio at batch 1 and throughput at batch 256, with the external flow-matching systems also shown after a round trip through our codec.

Table 5: **Ours against open systems on LibriSpeech-PC.** Data: training hours. NFE per utterance (non-autoregressive) or per block (ours); the flow-matching systems take two per step. The round trips through our codec are the fair reference for SIM. F5-TTS at 4 steps is not intelligible (21.5 % WER). TTFA and throughput on one B300, end to end (section D); VoXtream2's batched throughput is an upper bound.

						batch 1	batch 256
	Data (h)	NFE	WER (%) ↓	SIM ↑	PQ ↑	TTFA (ms) ↓	audio s / s ↑
References							
Human speech		–	2.61	0.695	6.50	–	–
through our codec		–	3.12	0.559	6.39	–	–
Non-autoregressive flow matching							
F5-TTS v1, 32 steps	95k	64	1.91	0.675	6.39	263	91
through our codec		64	2.14	0.554	6.50		
F5-TTS v1, 8 steps	95k	16	2.50	0.653	6.22	71	354
E2-TTS, 32 steps	95k	64	2.39	0.714	6.17	281	79
through our codec		64	2.63	0.575	6.36		
Streaming codec language model							
VoXtream2	40k	AR	3.32	0.659	6.30	97	369
Ours, block 8							
Blockwise flow matching, 16 steps	960	32	3.19	0.313	6.63	38	467
MeanFlow, 1 step	960	1	4.95	0.330	6.39	18	1593
MeanFlow, 2 steps	960	2	2.78	0.343	6.62	20	1472
MeanFlow, 3 steps	960	3	2.80	0.344	6.66	21	1368
MeanFlow + critic, 1 step	960	1	3.19	0.366	6.57	18	1593

Table 6 gives every timing cell behind the TTFA column of table 5 and fig. 8. All systems run on the same GPU with a 5 s prompt and a 15 s target, three warm-up runs and five timed runs, reporting the median. Throughput counts generated audio only; F5-TTS and E2-TTS denoise the prompt together with the target, so their wall clock covers 20 s of audio to deliver 15. VoXtream2 is held to exactly 15 s of audio with its end token disabled, as the other systems are by construction.

Table 6: **Timing on one B300.** Time to first audio end to end and generated audio seconds per wall-clock second, medians of five runs. NFE per step: network function evaluations per denoiser step (our guidance is one NFE; the flow-matching systems and VoXtream2 run a conditional and an unconditional pass). †: non-streaming, so the first audio arrives when the utterance is done. ‡: VoXtream2's batched mode synthesises identical utterances with early stopping disabled, an upper bound on its throughput.

			batch 1		batch 16	
system	steps	NFE / step	TTFA (ms)	audio s / s	TTFA (ms)	audio s / s
Ours, block 8	1	1	18	172	37	1058
	2	1	20	125	39	888
	3	1	21	99	40	765
	8	1	29	48	50	454
Ours, block 16	1	1	17	304	38	1416
	2	1	20	224	40	1249
	3	1	21	180	43	1112
	8	1	28	93	53	727
Ours, block 4	1	1	18	96	36	755
	2	1	19	67	37	601
	3	1	21	52	39	497
	8	1	28	24	49	268
Ours, block 32	1	1	17	447	42	1685
	2	1	19	356	44	1558
	3	1	20	296	46	1448
	8	1	28	163	57	1074
F5-TTS v1 (compiled)	4	2	39†	389	351†	683
	8	2	71†	212	679†	354
	16	2	135†	111	1334†	180
	32	2	263†	57	2645†	91
E2-TTS (compiled)	4	2	39†	381	404†	594
	8	2	74†	203	784†	306
	16	2	143†	105	1548†	155
	32	2	281†	53	3079†	78
VoXtream2 (compiled)	AR	2	97	5.2	103‡	76‡

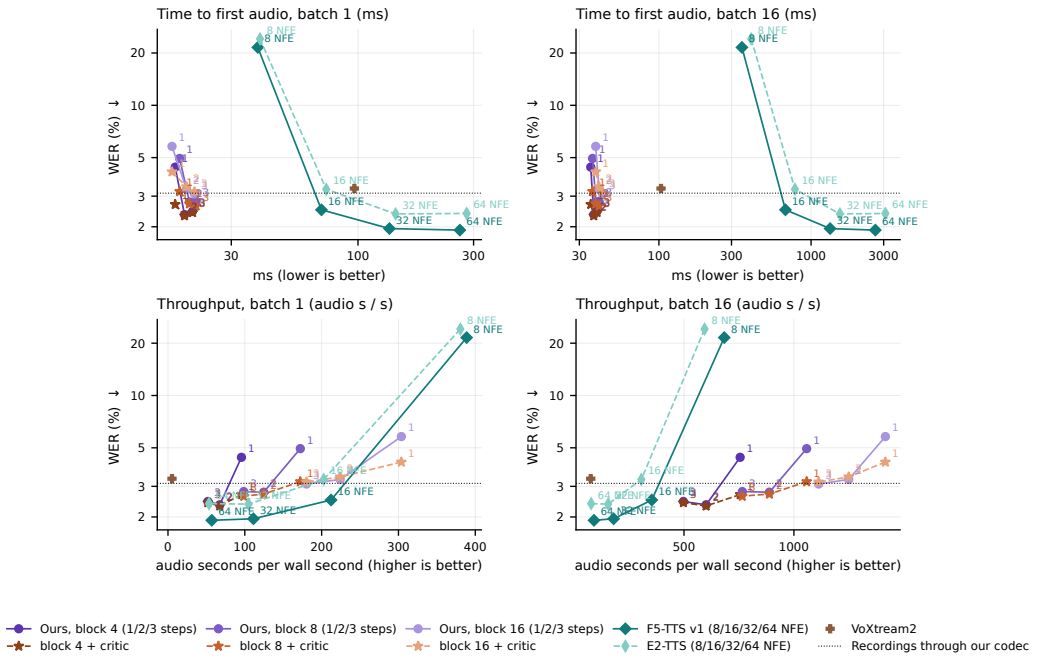

Figure 10: Intelligibility against time to first audio (log) and throughput at batch 1 and batch 16 on one B300: ours at blocks 4, 8 and 16 with one, two and three NFE (solid) and after the critic (dashed, stars); F5-TTS at 4 to 32 steps (8 to 64 NFE); E2-TTS; VoXtream2. The dotted line is the WER of human speech through our codec.