TAMEDFLOW: TRAJECTORY-AWARE DISTILLATION FOR TEACHER-ASSISTED DIFFUSION SAMPLING

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion Transformers (DiTs) deliver high-fidelity image and video generation, but inference requires many sequential model evaluations. Two acceleration strategies are increasingly important: model compression makes each evaluation cheaper, while adaptive inference avoids selected teacher computation through reuse, forecasting, or speculative execution. Their combination, however, is brittle: directly applying the evaluated adaptive methods to a compressed student can degrade quality, an observed failure mode we call the **Combination Gap**. We link this gap to a mismatch between pointwise distillation and short-horizon execution: a student trained at teacher states need not remain reliable when its own predictions are chained. We propose **TamedFlow**, which trains a compressed DiT for this deployed role through cross-architecture feature alignment, a near-identity Velocity Alignment Head, and randomized rollout supervision. At inference, a lightweight Learnable Trajectory Estimator routes completed drafts: accepted drafts bypass the teacher, while rejected drafts roll back and receive one teacher rescue step. Distilling DiT-XL/2 into DiT-B/2, TamedFlow reaches FID 2.58 at $2.25\times$ measured speedup over Euler-50 on ImageNet 256×256. Ablations support the roles of trajectory supervision and teacher rescue, and evaluations on ImageNet 512×512, MS-COCO, and Latte/FaceForensics extend the empirical scope.

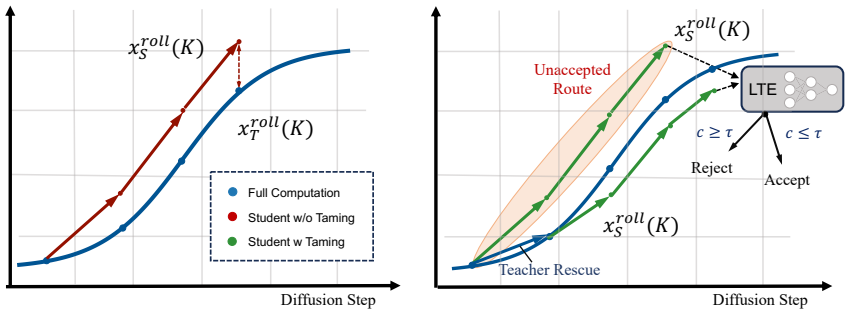
1 INTRODUCTION

Diffusion Transformers (DiTs) have become a leading backbone for high-fidelity visual generation (Peebles & Xie, 2023), yet iterative denoising imposes high inference latency. Each sample requires dozens of sequential forward passes, with costs scaling rapidly with model parameters, resolution, and sequence length. Existing acceleration strategies attack complementary aspects of this cost: step-reduction methods shorten the solver schedule (Song et al., 2023; Luo et al., 2023; Geng et al., 2026; Kim et al., 2025); parameter compression lowers per-step computation; and adaptive inference reuses or forecasts intermediate activations (Ma et al., 2024b; Selvaraju et al., 2024; Liu et al., 2025a;b). Combining parameter compression with adaptive execution is intuitively appealing, promising cheaper evaluations while skipping redundant computation. All headline experiments below use fixed-step Euler integration to evaluate compression and adaptive execution under a unified solver convention.

In practice, this composition can break down. In our controlled diagnostic (Appendix Table 11), attaching the evaluated adaptive reuse methods directly to a compressed student degrades sample quality, inflating ImageNet 256×256 FID from 5.56 to between 10.40 and 13.18. We term this failure mode the **Combination Gap**. This result exposes an operational mismatch: a student trained for pointwise agreement need not remain reliable after its own predictions alter the states encountered later. Rather than applying internal reuse unchanged to the compact model, we let the student generate short trajectory drafts and reserve teacher intervention for drafts predicted to deviate.

Chaining student predictions into drafts, however, introduces compounding integration drift. Proposition 3.1 characterizes this through an Euler rollout bound: spatial sensitivity can amplify local student–teacher velocity mismatch over sequential steps. To stabilize drafting against this amplification, we introduce **TamedFlow**, combining trajectory-aware distillation with selective teacher rescue (Figure 1). During training, cross-architecture feature alignment (AlignNet) grounds student representations in teacher layers; a near-identity Velocity Alignment Head (VAH) absorbs residual

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.



(a) Trajectory Drift (b) Trajectory Taming

Figure 1: **The Combination Gap and TamedFlow.** Directly applying adaptive reuse to a parameter-compressed student severely degrades quality in our diagnostic. TamedFlow instead trains the compressed student for short-horizon trajectory agreement. A 4M-parameter LTE routes drafts: accepted drafts bypass the teacher; rejected drafts roll back for one teacher rescue step.

velocity errors; and randomized K-step rollout supervision ($K \sim \mathcal{U}(1, 5)$) penalizes both intermediate mismatch and accumulated endpoint drift. A two-stage curriculum introduces the multi-step objectives only after feature alignment.

At inference, TamedFlow uses an asymmetric execution policy. A compact Learnable Trajectory Estimator (LTE, 4M parameters) scores completed drafts from pooled student tokens and timestep embeddings without teacher access. If confidence exceeds threshold τ, the draft is accepted, advancing K steps while bypassing the teacher. If rejected, the sampler rolls back to the pre-draft state and advances one step with teacher rescue. Varying τ selects among measured operating points between throughput and fidelity without sequential teacher verification.

Distilling DiT-XL/2 into a 130M DiT-B/2 student on ImageNet 256×256, TamedFlow attains FID 2.58 at $2.25\times$ measured speedup (0.464 s vs. 1.044 s) over the Euler-50 teacher. At comparable latency, TamedFlow improves FID over the evaluated training-free adaptive baselines (e.g., FoCa at FID 2.65, 0.466 s). Controlled ablations support the contributions of trajectory supervision and teacher rescue, while evaluations on ImageNet 512×512, MS-COCO, and Latte video DiTs extend the empirical scope. In summary, our contributions are:

- **Diagnostic and analysis of the Combination Gap.** We document the failure of applying the evaluated adaptive reuse methods to a compressed DiT, and derive an Euler rollout bound characterizing how sequential drafting can amplify local velocity errors.
- **Trajectory-stabilized speculative distillation.** We introduce TamedFlow, uniting feature alignment, residual velocity correction, and randomized rollout supervision to optimize a compressed DiT for autonomous multi-step drafting.
- **Learned routing and deployment evaluation.** We propose an asymmetric execution policy via a 4M-parameter LTE to accept reliable drafts or trigger teacher rescue, evaluating quality–latency trade-offs across resolutions, text conditioning, and video alongside physical deployment costs.

2 Related Work

Prior work on DiT acceleration reduces the number of solver evaluations, the cost of each evaluation, or the teacher computation retained during adaptive execution. TamedFlow studies their interface: whether a compressed student can reliably replace short segments of a teacher trajectory.

2.1 Step Count Reduction in Diffusion and Flow Models

Step reduction lowers the number of solver evaluations. Higher-order solvers improve numerical efficiency for a fixed model (Song et al., 2021; Lu et al., 2025), while distillation changes the model: progressive distillation shortens the schedule (Salimans & Ho, 2022), consistency models match predictions across noise levels (Song et al., 2023; Luo et al., 2023), MeanFlow and Model-

Guidance alter the target or absorb guidance (Geng et al., 2026; Tang et al., 2025). Other formulations align distributions or scores, distill autoregressively, transfer trajectories, or train few-step students adversarially (Yin et al., 2024; Zhou et al., 2025; Kim et al., 2025; Luo et al., 2025; Sauer et al., 2024). These methods optimize a sparse standalone sampler deployed entirely on its own. TamedFlow instead retains a multi-step teacher as a reliable safeguard and uses the student to propose short trajectory segments whose acceptance is decided adaptively online.

A related line studies trajectory geometry. Rectified flow and InstaFlow construct straighter paths (Liu et al., 2023); interpolant transformers unify diffusion and flow parametrizations (Ma et al., 2024a); and guided path sampling corrects trajectories during integration (Li et al., 2026). TCD and Hyper-SD learn consistency or trajectory segments for few-step sampling (Zheng et al., 2024; Ren et al., 2024). TamedFlow keeps the teacher path as the reference, trains a cross-architecture student for selectively accepted rollouts, and retains teacher rescue in the deployed sampler.

2.2 Per-Step Compute Acceleration and Feature Caching

Per-step methods keep the schedule while avoiding full recomputation. Feature-reuse methods cache activations across timesteps or blocks (Ma et al., 2024b; Chen et al., 2024; Selvaraju et al., 2024; Zou et al., 2025; 2026b; Wimbauer et al., 2024); forecasting methods extrapolate future features (Liu et al., 2025a;b; Cai et al., 2026; Huang et al., 2025); and learned schedulers select policies per sample (Zhao et al., 2026). These methods exploit local regularity in intermediate computations. Our same-student diagnostic shows that their evaluated configurations do not transfer directly to the compressed student, but does not isolate a universal caching failure mechanism.

Most reuse methods leave the model unchanged and decide which internal computations can be skipped. TamedFlow instead trains the student under rollout supervision and learns whether an entire student-generated segment can advance without a teacher call. DisCa (Zou et al., 2026a) also studies incompatibility between step-distilled video DiTs and feature caches, but repairs the cache side with a predictor and Restricted MeanFlow schedule. TamedFlow acts on the draft student instead.

2.3 Speculative Verification in Generative Paradigms

Speculative decoding verifies proposed tokens with a target model (Leviathan et al., 2023); causal masking permits parallel verification and exact correction. Diffusion lacks this structure because each teacher state depends on the preceding prediction. Replaying a K-step draft exactly would require sequential teacher evaluations and remove much of the saving. SpecDiff (Pan et al., 2026) studies self-speculative diffusion under this constraint. TamedFlow replaces exact verification with learned trajectory estimation. The LTE scores student-internal features after a draft using rollout agreement as its training target. Accepted drafts avoid the teacher; rejected drafts return to the pre-draft state for one teacher step. Routing is therefore inexpensive and online, but approximate and not distribution-preserving by construction.

3 Method

3.1 Preliminaries: Diffusion Models, DiT, and Flow Matching

Diffusion models (Ho et al., 2020) learn to reverse a gradual noising process, so generation requires a sequence of state-dependent model evaluations. A Diffusion Transformer (DiT) (Peebles & Xie, 2023) implements the denoiser with a transformer over latent patches and injects timestep and condition information through adaptive layer normalization. Our method is expressed in velocity form because both its trajectory loss and its stability analysis act on the vector field followed during sampling.

Flow matching (Lipman et al., 2023) casts generation as a probability-path ODE. With $x_0 \sim \mathcal{N}(0, I)$, $x_1 \sim p_{\text{data}}$, and the linear interpolant $x_t = (1-t)x_0 + tx_1$, the conditional velocity $x_1 - x_0$ supplies the regression target:

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{t,x_0,x_1}\left[\left\|v_\theta(x_t,t) - (x_1 - x_0)\right\|_2^2\right], \qquad t \sim \mathcal{U}[0,1]. \tag{1}$$

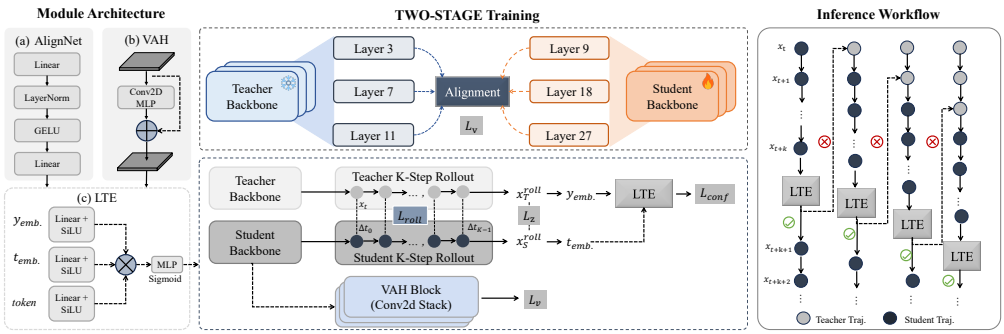


Figure 2: TamedFlow overview. Left: AlignNet, VAH, and LTE attached to the student backbone. Middle: two-stage curriculum. Stage 1 is feature alignment and Stage 2 is rollout supervision. Right: inference routes drafts via the LTE; accepted drafts bypass the teacher, rejected drafts roll back and take one teacher step.

We use forward flow time, from noise at $t = 0$ to data at $t = 1$. The main experiments use the stated Euler schedules; the Euler discretization below is an analysis device and the rollout rule used to define the trajectory objectives.

3.2 Spatial Error Amplification and Rollout Stability

Pointwise velocity matching constrains the student at sampled states but not the states reached after its own predictions are chained. The following proposition separates the local mismatch that initiates drift from the spatial sensitivity that amplifies it. It applies to sequential Euler drafts and does not analyze internal feature caching or forecasting.

Proposition 3.1 (K-Step Trajectory Alignment Bound). *Let $\tilde{v}_S \equiv v_S^{\text{aligned}}$ denote the deployed student velocity field, and let $x_S^{\text{roll}}(K)$ and $x_T^{\text{roll}}(K)$ be the aligned-student and teacher endpoints after K sequential Euler steps from state x_t with fixed step size h. With time arguments suppressed at the common $t_k = t + kh$, define the baseline velocity mismatch $\delta_v^{(k)} \triangleq \tilde{v}_S(x_T^{(k)}) - v_T(x_T^{(k)})$, with $\|\delta_v\|_\infty \triangleq \max_k \|\delta_v^{(k)}\|_2$. Let $L_T \triangleq \sup_{x,t} \|J_{v_T}(x,t)\|_2$ and $\Delta J \triangleq \sup_{x,t} \|J_{\tilde{v}_S}(x,t) - J_{v_T}(x,t)\|_2$. Assuming both velocity fields are continuously differentiable on a region containing each segment between paired student and teacher states and $C \triangleq L_T + \Delta J < \infty$, the spatial deviation satisfies the discrete Grönwall upper bound:*

$$\|x_S^{\text{roll}}(K) - x_T^{\text{roll}}(K)\|_2 \le \frac{\|\delta_v\|_\infty}{C}\big((1+hC)^K - 1\big), \tag{2}$$

where $h > 0$ is the fixed Euler integration step size; for $C = 0$, the right-hand side is interpreted as $Kh\|\delta_v\|_\infty$. For $KhC \ll 1$, this implies the few-step expansion $Kh\|\delta_v\|_\infty + \mathcal{O}(K^2h^2C\|\delta_v\|_\infty)$.

The recurrence $\|e_{k+1}\|_2 \le (1+hC)\|e_k\|_2 + h\|\delta_v\|_\infty$, unrolled from $e_0 = 0$, gives Eq. equation 2; Appendix F gives the proof. The implication for method design is limited but useful: reducing one-step mismatch is insufficient when the same error is repeatedly propagated, so training should also expose the student to short rollouts. The bound neither establishes a universal quadratic error law nor guarantees that our objectives minimize C.

3.3 Stagewise Training

Figure 2 summarizes AlignNet for feature alignment, VAH for residual velocity correction, and LTE for draft scoring. A two-stage curriculum separates feature initialization from rollout supervision.

Setup and state extraction. Let v_T and v_S denote the teacher and student velocity networks. Pointwise distillation uses $\mathcal{L}_v = \mathbb{E}[D^{-1}\|v_S(x_t,t) - v_T(x_t,t)\|_2^2]$, where D is the latent dimension; Stage 2 applies this loss to the aligned student velocity defined below. The student backbone also exposes hidden tokens from its ordinary forward pass, which the LTE later uses for routing without an additional backbone evaluation.

Stage 1: Feature alignment. Stage 1 freezes VAH and trains the backbone with velocity and intermediate-feature matching. AlignNet uses 2-layer MLP projectors to pair student layers $\{3, 7, 11\}$ with teacher layers $\{9, 18, 27\}$, spanning early through late representations.

$$\mathcal{L}_{\text{align}} = \frac{1}{3} \sum_{l \in \{3,7,11\}} \frac{\|A_l(f_S^l) - f_T^{\pi(l)}\|_2^2}{\|f_T^{\pi(l)}\|_2^2}. \tag{3}$$

Here $\pi(3) = 9$, $\pi(7) = 18$, and $\pi(11) = 27$. The Stage 1 objective $\mathcal{L}_{\text{stage1}} = \mathcal{L}_v + \lambda_{\text{align}}\mathcal{L}_{\text{align}}$ initializes a student whose intermediate representation and velocity prediction are both anchored to the teacher before free-running supervision begins.

Stage 2: Rollout supervision. After Stage 1 stabilizes, we activate rollout-level supervision. To learn a residual velocity correction, we unfreeze the **Velocity Alignment Head (VAH)**, a convolutional residual head initialized with a small output correction, so $v_S^{\text{aligned}} \approx v_S$:

$$v_S^{\text{aligned}} = v_S + \text{VAH}(v_S; \phi). \tag{4}$$

Stage 2 trains the model for short-horizon use. We compare aligned student and teacher velocities at teacher states and separately penalize the endpoints obtained when each model follows its own rollout:

$$\mathcal{L}_{\text{stage2}} = \mathcal{L}_v + \lambda_{\text{roll}}\mathcal{L}_{\text{roll}}(K) + \lambda_z\mathcal{L}_z + \lambda_{\text{conf}}\mathcal{L}_{\text{conf}}, \tag{5}$$

where $\mathcal{L}_{\text{roll}}(K) = \frac{1}{KD}\sum_{k=0}^{K-1} \|v_S^{\text{aligned}}(x_T^{(k)}, t{+}k\Delta t) - v_T(x_T^{(k)}, t{+}k\Delta t)\|_2^2$ is teacher-state velocity anchoring, whereas $\mathcal{L}_z = D^{-1}\|x_S^{\text{roll}}(K) - x_T^{\text{roll}}(K)\|_2^2$ is the free-running endpoint loss obtained by rolling each field from the same start state on its own states. The endpoint loss backpropagates through the unrolled student steps, while training samples K uniformly from $\{1, 2, 3, 4, 5\}$ so the model is not specialized to one horizon. For any fixed teacher-state rollout, $\|\delta_v\|_\infty \le \sqrt{KD\mathcal{L}_{\text{roll}}}$; this links the empirical loss to the forcing term in Proposition 3.1, but is not a uniform guarantee from the average training objective. The **LTE** is trained with

$$\mathcal{L}_{\text{conf}} = \text{MSE}\big(\text{LTE}(\text{sg}[\bar{f}_S^{\text{last}}, e_{t_{K-1}}, e_y]), \text{sg}[c^\star]\big), \tag{6}$$

where $\bar{f}_S^{\text{last}}$ denotes mean-pooled hidden tokens from the last student forward in the draft and $\text{sg}[\cdot]$ denotes stop-gradient. Thus the LTE learns from the same type of signal available online, while its loss does not perturb the student backbone.

3.4 Self-Speculative Execution via Confidence Routing

For a K-step draft, the target score $c^\star$ combines endpoint rollout deviation with the velocity error at the shared start state:

$$c^\star = \exp\Big(-\alpha\frac{\|x_S^{\text{roll}}(K) - x_T^{\text{roll}}(K)\|_2^2}{DT_z} - (1-\alpha)\frac{\|v_S^{\text{aligned}}(x_t, t) - v_T(x_t, t)\|_2^2}{DT_v}\Big), \tag{7}$$

with latent dimension D and temperatures T_z, T_v. The score decreases as either discrepancy grows; it is a regression target, not a calibrated probability that a draft preserves sample quality. At inference the LTE predicts this score from student features alone.

LTE inference. For a clipped draft of length m, the LTE uses mean-pooled tokens and the timestep embedding from the last drafting forward at t_{i+m-1}, together with the condition representation:

$$h_{\text{LTE}} = \sigma(g_f)W_1\bar{f}_S + \sigma(g_t)W_2 e_{t_{i+m-1}} + \sigma(g_y)W_3 e_y, \qquad c = \sigma\big(w^\top \text{MLP}(h_{\text{LTE}}) + b\big). \tag{8}$$

The LTE has no explicit horizon input, so its score is an empirical routing signal rather than a horizon-specific error guarantee.

Routing. The student drafts K steps, clipped at the schedule boundary. If $c \ge \tau$, the sampler accepts the entire draft and bypasses the teacher; otherwise it discards the tentative states, restores the pre-draft state, and advances one step with the teacher. Appendix H gives the full algorithm.

Table 1: ImageNet 256 × 256 class-conditional generation with DiT-XL/2 using fixed-step Euler.

Method	Latency (s) ↓	Speedup ↑	FLOPs (T) ↓	FID ↓	IS ↑
Euler-50 Teacher	*1.044*	*1.00×*	*23.74*	*2.35*	*237.62*
Pure Student	0.180	5.80×	2.30	5.56 (+3.21)	195.07 (-42.55)
Full-Scale Few-Step Distillation					
Progressive Distill. (step=4) (Salimans & Ho, 2022)	0.068	15.35×	0.474	10.25 (+7.90)	181.58 (-56.04)
LCM (step=4) (Luo et al., 2023)	0.072	14.50×	0.476	6.81 (+4.46)	202.73 (-34.89)
ARD (step=4) (Kim et al., 2025) ($N = 6$)	0.074	14.11×	0.480	4.32 (+1.97)	209.03 (-28.59)
1.5×–2.0× Speedup Range					
Euler-25	0.558	1.87×	11.87	3.21 (+0.86)	229.91 (-7.71)
Δ-DiT (Chen et al., 2024) ($N = 2$)	0.684	1.53×	18.04	2.78 (+0.43)	223.47 (-14.15)
TaylorSeer (Liu et al., 2025a) ($N = 2$)	0.630	1.66×	12.35	2.47 (+0.12)	230.21 (-7.41)
FoCa (Zheng et al., 2026) ($N = 2$)	0.634	1.65×	12.35	2.37 (+0.02)	233.70 (-3.92)
SpecDiff (Pan et al., 2026)	0.644	1.62×	14.58	2.39 (+0.04)	235.18 (-2.44)
TamedFlow ($\tau = 0.85$)	**0.602**	**1.73**×	**8.53**	**2.33** (-0.02)	**236.15** (-1.47)
2.0×–2.5× Speedup Range					
Euler-20	0.442	2.36×	9.49	3.87 (+1.52)	219.40 (-18.22)
FORA (Selvaraju et al., 2024) ($N = 3$)	0.482	2.17×	8.58	3.65 (+1.30)	228.14 (-9.48)
TaylorSeer (Liu et al., 2025a) ($N = 3$)	0.477	2.19×	8.56	2.73 (+0.38)	229.36 (-8.26)
FoCa (Zheng et al., 2026) ($N = 3$)	0.466	2.24×	8.56	2.65 (+0.30)	230.56 (-7.06)
SpeCa (Liu et al., 2025b)	0.504	2.07×	**4.76**	2.78 (+0.43)	229.74 (-7.88)
TamedFlow ($\tau = 0.80$)	**0.464**	**2.25**×	6.83	**2.58** (+0.23)	**233.43** (-4.19)

4 Experiments

4.1 Experimental Setup

Models and datasets. The primary setting distills a 675M DiT-XL/2 teacher $\mathcal{T}_\phi$ (Peebles & Xie, 2023) into a 130M DiT-B/2 student $\mathcal{B}_\theta$ on class-conditional ImageNet at 256×256. We evaluate resolution scaling on ImageNet 512×512, text conditioning on MS-COCO, and video generation with Latte on FaceForensics. For COCO, a shared 19.9M text-conditioning adapter maps 77-token CLIP ViT-L/14 features into the DiT conditioning space; the teacher and student use this shared adapter while the ImageNet-trained VAH and LTE remain fixed. For Latte, the student, AlignNet, VAH, and LTE are retrained for the evaluated video-DiT setting. Complete training schedules, solver settings, and adaptation protocols are provided in Appendices A and E.

Metrics and reporting. On ImageNet, we report FID-50K and Inception Score (IS); on MS-COCO, FID-30K and CLIP-Score computed with OpenAI CLIP ViT-L/14; and on FaceForensics, FVD and video Inception Score. Image and text-to-image wall-clock latency and speedup are measured on a single A800-80GB GPU at batch size one and averaged over 10 runs. All baseline rows are measured in our local implementations under the same reporting protocol. Speedup is normalized to the corresponding teacher, and the LTE accept rate is the fraction of routing events that retain the student draft. FLOPs count backbone evaluations, whereas latency also includes VAH, LTE, rollback, and solver overhead.

Baselines. We compare against schedule-reduction references (Euler-25 and Euler-20), full-scale few-step distillation methods (Progressive Distillation (Salimans & Ho, 2022), LCM (Luo et al., 2023), and ARD (Kim et al., 2025)), and adaptive or speculative methods (Δ-DiT (Chen et al., 2024), TaylorSeer (Liu et al., 2025a), FoCa (Zheng et al., 2026), FORA (Selvaraju et al., 2024), SpeCa (Liu et al., 2025b), and SpecDiff (Pan et al., 2026)). Adaptive and speculative baselines are re-implemented in the matched DiT-XL/2 pipeline and measured locally under the same latency protocol. Few-step distillation baselines use the full teacher architecture without parameter compression. Appendix B details the comparison scope and accounting assumptions.

4.2 Main Results

Quality–latency operating points. Table 1 compares complete systems across speedup regimes. Four-step distillation baselines attain aggressive $14\times$–$15\times$ acceleration at the cost of severe FID degradation. TamedFlow instead targets moderate acceleration with tight quality retention: at



Figure 3: Qualitative comparison on ImageNet 256 × 256.

$\tau = 0.85$, it matches the Euler-50 teacher within 0.02 FID (2.33 vs. 2.35) at 1.73× speedup; at default $\tau = 0.80$, it reaches FID 2.58 at 2.25× speedup (0.464 s).

In the 2.0× to 2.5× range under nearly matched latencies around 0.46 to 0.48 seconds, TamedFlow achieves better quality than FoCa at FID 2.65 and TaylorSeer at FID 2.73, alongside a higher Inception Score. This represents a quality advantage at matched latency rather than a throughput claim from millisecond-scale differences. Although SpeCa reports lower backbone FLOPs of 4.76 T, its FID is 0.20 worse, and backbone FLOPs omit runtime overheads like rollback, VAH, and LTE that are fully captured by measured latency. Thus, TamedFlow's edge lies in its empirical Pareto trade-off for the deployed system, not a FLOPs-only claim.

Qualitative study. Figure 3 compares generated samples under matched speedup (2.1×–2.25×). TamedFlow preserves fine textures and object boundaries more cleanly across diverse classes. For example, it avoids the over-smoothed fur patterns on *Tabby* and *Boxer* seen in competing adaptive methods, and retains clear structural separation between the metal hoops and wood grain on *Barrel*. These visual examples align with aggregate FID trends, while the following ablations isolate the underlying mechanisms.

4.3 Component Ablation

Table 2 isolates the contribution of each architectural module and training objective at draft length $K = 3$ and routing threshold $\tau = 0.80$.

Table 2: Ablation studies on ImageNet 256 × 256 at $K = 3$ and $\tau = 0.80$. Speedup is relative to the 50-step teacher baseline.

Study	Variant	Accept % ↑	Speedup ↑	FID ↓	IS ↑
Component	Full TamedFlow	63.6	2.25×	**2.58**	**233.43**
	w/o AlignNet	46.2	1.87×	2.95	211.34
	w/o VAH	39.3	1.63×	3.23	205.71
	w/o LTE	–	5.80×	5.56	195.07
Training	Full TamedFlow	63.6	2.25×	**2.58**	**233.43**
	w/o $\mathcal{L}_{\text{roll}}$	54.2	2.02×	2.71	230.15
	w/o $\mathcal{L}_z$ warmup	46.2	1.81×	3.29	203.50
	w/o $\mathcal{L}_z$	28.1	1.45×	4.15	198.80

Modules and training objectives. AlignNet initializes student representations from intermediate teacher layers; removing it drops acceptance to 46.2% and raises FID to 2.95 due to suboptimal

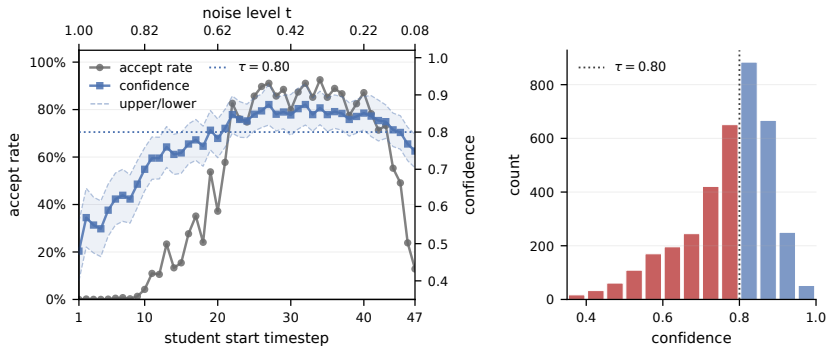


Figure 4: **LTE routing behavior across generation.** Left: mean routing confidence across timesteps, showing a drop near $t = 0.22$ as generation nears the data manifold. Right: score distributions of accepted and rejected drafts under $\tau = 0.80$.

representation grounding. Disabling the VAH reduces acceptance to 39.3% and increases FID to 3.23, reflecting its role in absorbing local residual velocity mismatch (without claiming an explicit theoretical reduction in ΔJ). Without the LTE, unrouted student execution reaches 5.80× speedup but degrades severely to FID 5.56 in the absence of corrective recovery. For training losses, omitting rollout supervision $\mathcal{L}_{\text{roll}}$ weakens intermediate velocity consistency and increases FID to 2.71. Removing the endpoint loss $\mathcal{L}_z$ has the most severe impact, dropping acceptance to 28.1% and inflating FID to 4.15; its unrolled multi-step gradient directly penalizes compounding trajectory drift during autonomous rollouts. Removing the 10-epoch warmup on $\mathcal{L}_z$ yields FID 3.29 due to early optimization shocks before feature representations stabilize.

Teacher-rescue counterfactual. To test whether teacher recovery is necessary, we evaluate a counterfactual where rejected drafts roll back and advance with the student itself (Student+VAH; Appendix Table 8). This policy severely degrades FID from 2.58 to 7.12, performing even worse than the unrouted dense student baseline at 5.56. Retrying with the same approximate student vector field fails to correct the local integration drift that triggered the rejection in the first place, demonstrating that switching back to the high-capacity teacher vector field is essential for error recovery at the evaluated operating point.

4.4 Routing Dynamics and Sensitivity

Draft length K and threshold τ balance speedup against fidelity (Appendix Figure 10). Speedup peaks at $K = 3$ because longer drafts compound drift and increase rollback costs upon rejection. As shown in Figure 4, LTE confidence remains high in the mid-noise regime but drops near $t = 0.22$, invoking more teacher rescues as generation nears the data manifold. The distribution shows clear operational separation between accepted and rejected drafts at the chosen operating point $\tau = 0.80$.

At $K = 3$ and $\tau = 0.80$, 14 of 22 routing events are accepted, allowing student rollouts to cover 42 of the 50 integration steps. Detailed component profiling reveals that teacher rescue accounts for 71.9% of wall-clock latency, whereas student rollouts take 14.9%, rejections with state rollback consume 12.7%, and LTE confidence scoring incurs only 0.5%. The lightweight estimator itself introduces negligible computational latency, but routing accuracy remains paramount: false acceptances allow subtle draft errors to compound and degrade fidelity, while false rejections trigger expensive teacher interventions that erode throughput. This concrete latency breakdown also explains why measured end-to-end speedup does not strictly match backbone FLOPs, as runtime costs from state rollback, synchronization, and auxiliary modules are omitted from nominal FLOP totals.

4.5 Cross-Setting Evaluation

On MS-COCO, the teacher and student share a trained 19.9M text adapter while the ImageNet-trained VAH and LTE remain frozen. As reported in Table 3, TamedFlow reaches FID 14.86 and CLIP-Score 29.6 at 1.96× speedup at $\tau = 0.80$, and 2.14× speedup with FID 15.52 at $\tau = 0.75$, outperforming

Table 3: MS-COCO 30K 256×256 text-to-image generation using fixed-step Euler. TaylorSeer, SpeCa, and FoCa are re-implemented within our DiT-XL/2 pipeline; N denotes each method's acceleration setting rather than TamedFlow's draft horizon K.

Method	**Latency (s)** ↓	**Speedup** ↑	**FID** ↓	**CLIP-S** ↑
Teacher Euler-50	*1.184*	*1.00×*	*13.54*	*30.2*
Teacher Euler-25	*0.622*	*1.90×*	*16.84*	*28.9*
Pure Student	0.203	5.83×	19.30	27.8
TaylorSeer ($N = 3$)	0.580	2.04×	15.47	29.1
SpeCa	0.561	2.11×	16.63	28.4
FoCa ($N = 3$)	0.565	2.09×	16.27	28.5
TamedFlow ($\tau = 0.80$)	0.603	1.96×	**14.86**	**29.6**
TamedFlow ($\tau = 0.75$)	**0.551**	**2.14**×	15.52	29.2

SpeCa and FoCa at comparable latency. This supports transfer of the frozen alignment modules under condition adaptation rather than unadapted zero-shot deployment. Tightening τ improves FID by 0.66 and CLIP-Score by 0.4 for 52 ms latency, verifying that fidelity recovery stems from teacher intervention rather than student capacity.

We further evaluate spatial and temporal scaling. On ImageNet 512×512, expanding the latent grid fourfold under the same backbone, TamedFlow achieves FID 3.39 at $2.41\times$ speedup in 1.594 s, outperforming FoCa at FID 3.47 with $2.36\times$ speedup as reported in Appendix C. On the Latte video DiT retrained on FaceForensics at 256×256 resolution across 16 frames in Table 4, TamedFlow attains FVD 225.10 and IS 68.90 at $2.22\times$ speedup, surpassing FoCa at FVD 248.30 and TaylorSeer at FVD 253.10. These results confirm that the draft and rescue mechanism transfers reliably across resolutions, conditions, and video modalities.

Table 4: Latte video-DiT transfer on FaceForensics (256×256, 16 frames).

Method	**Speedup**	**FVD** ↓	**IS** ↑
Teacher (Latte-XL)	1.00×	208.50	71.20
FoCa	2.06×	248.30	66.80
TaylorSeer	2.10×	253.10	65.90
TamedFlow	**2.22**×	**225.10**	**68.90**

Deployment cost. TamedFlow requires offline training and maintains the teacher, student, VAH, and LTE co-resident in memory, increasing peak allocated VRAM from 2.570 to 3.075 GB ($\sim$0.5 GB overhead). This footprint reflects the deliberate choice to retain a high-capacity teacher for selective rescue rather than discarding it. Headline latency is benchmarked for batch-one latent integration without VAE decoding, characterizing interactive generation rather than batched serving throughput. Teacher offloading and weight quantization represent orthogonal opportunities to mitigate co-residency overhead in production.

5 Conclusion

In this work, we introduced TamedFlow, an acceleration framework that reconciles parameter compression with adaptive execution for Diffusion Transformers. By training a compact student for short-horizon trajectory agreement and employing asymmetric routing with selective teacher rescue, TamedFlow achieves FID 2.58 at $2.25\times$ speedup over Euler-50 on ImageNet 256×256, with transfer evaluated across resolutions, text conditioning, and video. Our findings show that pointwise alignment can leave compounding trajectory divergence during autonomous drafting, creating the Combination Gap. Controlled ablations support the roles of rollout supervision and teacher recovery at the reported operating point, suggesting that compression and adaptive inference can be made compatible when the compact model is trained as a multi-step drafter and the sampler retains a recovery path back to the teacher.

REFERENCES

Peiliang Cai, Jiacheng Liu, Haowen Xu, Xinyu Wang, Chang Zou, and Linfeng Zhang. Lesa: Learnable stage-aware predictors for diffusion model acceleration. *arXiv preprint arXiv:2602.20497*, 2026.

Pengtao Chen, Mingzhu Shen, Peng Ye, Jianjian Cao, Chongjun Tu, Christos-Savvas Bouganis, Yiren Zhao, and Tao Chen. δ-dit: A training-free acceleration method tailored for diffusion transformers, 2024. URL https://arxiv.org/abs/2406.01125.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *Advances in Neural Information Processing Systems*, 38:75460–75482, 2026.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Yushi Huang, Zining Wang, Ruihao Gong, Jing Liu, Xinjie Zhang, Jinyang Guo, Xianglong Liu, and Jun Zhang. Harmonica: Harmonizing training and inference for better feature caching in diffusion transformer acceleration. In *ICML*, 2025.

Yeongmin Kim, Sotiris Anagnostidis, Yuming Du, Edgar Schönfeld, Jonas Kohler, Markos Georgopoulos, Albert Pumarola, Ali Thabet, and Artsiom Sanakoyeu. Autoregressive distillation of diffusion transformers. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15745–15756. IEEE, 2025.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *International conference on machine learning*, pp. 19274–19286. PMLR, 2023.

Haosen Li, Wenshuo Chen, Shaofeng Liang, Lei Wang, Haozhe Jia, and Yutao Yue. Guided path sampling: Steering diffusion models back on track with principled path guidance. In *Companion Proceedings of the ACM Web Conference 2026*, pp. 1295–1300, 2026.

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

Jiacheng Liu, Chang Zou, Yuanhuiyi Lyu, Junjie Chen, and Linfeng Zhang. From reusing to forecasting: Accelerating diffusion models with taylorseers. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15853–15863. IEEE, 2025a.

Jiacheng Liu, Chang Zou, Yuanhuiyi Lyu, Fei Ren, Shaobo Wang, Kaixin Li, and Linfeng Zhang. Speca: Accelerating diffusion transformers with speculative feature caching. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 10024–10033, 2025b.

Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, et al. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In *The twelfth international conference on learning representations*, 2023.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *Machine Intelligence Research*, 22 (4):730–751, 2025.

Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023.

Yihong Luo, Tianyang Hu, Jiacheng Sun, Yujun Cai, and Jing Tang. Learning few-step diffusion models by trajectory distribution matching. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17719–17728. IEEE, 2025.

Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, pp. 23–40. Springer, 2024a.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. Deepcache: Accelerating diffusion models for free. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15762–15772. IEEE, 2024b.

Jiayi Pan, Jiaming Xu, Yongkang Zhou, and Guohao Dai. Specdiff: Accelerating diffusion model inference with self-speculation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 8233–8241, 2026.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

Yuxi Ren, Xin Xia, Yanzuo Lu, Jiacheng Zhang, Jie Wu, Pan Xie, Xing Wang, and Xuefeng Xiao. Hyper-sd: Trajectory segmented consistency model for efficient image synthesis. *Advances in neural information processing systems*, 37:117340–117362, 2024.

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.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision*, pp. 87–103. Springer, 2024.

Pratheba Selvaraju, Tianyu Ding, Tianyi Chen, Ilya Zharkov, and Luming Liang. Fora: Fast-forward caching in diffusion transformer acceleration. *arXiv preprint arXiv:2407.01425*, 2024.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=St1giarCHLP.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

Zhicong Tang, Jianmin Bao, Dong Chen, and Baining Guo. Diffusion models without classifier-free guidance. *arXiv preprint arXiv:2502.12154*, 2025.

Felix Wimbauer, Bichen Wu, Edgar Schoenfeld, Xiaoliang Dai, Ji Hou, Zijian He, Artsiom Sanakoyeu, Peizhao Zhang, Sam Tsai, Jonas Kohler, et al. Cache me if you can: Accelerating diffusion models through block caching. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6211–6220. IEEE, 2024.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6613–6623. IEEE, 2024.

Wangbo Zhao, Yizeng Han, Zhiwei Tang, Jiasheng Tang, Pengfei Zhou, Kai Wang, Bohan Zhuang, Zhangyang Wang, Fan Wang, and Yang You. RAPID^3: Tri-level reinforced acceleration policies for diffusion transformer. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=sQ0g6EkpF7.

Jianbin Zheng, Minghui Hu, Zhongyi Fan, Chaoyue Wang, Changxing Ding, Dacheng Tao, and Tat-Jen Cham. Trajectory consistency distillation: Improved latent consistency distillation by semi-linear consistency function with trajectory mapping. *arXiv preprint arXiv:2402.19159*, 2024.

Shikang Zheng, Liang Feng, Xinyu Wang, Qinming Zhou, Peiliang Cai, Chang Zou, Jiacheng Liu, Yuqi Lin, Junjie Chen, Yue Ma, et al. Forecast then calibrate: Feature caching as ode for efficient diffusion transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 13449–13457, 2026.

Mingyuan Zhou, Yi Gu, Huangjie Zheng, Liangchen Song, Guande He, Yizhe Zhang, Wenze Hu, and Yinfei Yang. Score distillation of flow matching models. *arXiv preprint arXiv:2509.25127*, 2025.

Chang Zou, Xuyang Liu, Ting Liu, Siteng Huang, and Linfeng Zhang. Accelerating diffusion transformers with token-wise feature caching. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=yYZbZGo4ei.

Chang Zou, Changlin Li, Songtao Liu, Zhao Zhong, Kailin Huang, and Linfeng Zhang. Disca: Accelerating video diffusion transformers with distillation-compatible learnable feature caching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4590–4601, June 2026a.

Chang Zou, Shikang Zheng, Evelyn Zhang, Runlin Guo, Haohang Xu, Zhengyi Shi, Conghui He, Xuming Hu, and Linfeng Zhang. Rethinking token-wise feature caching: Accelerating diffusion transformers with dual feature caching. *IEEE Transactions on Image Processing*, 2026b.

Appendix A Training Hyperparameters

Table 5 summarizes the training hyperparameters for the two stages of TamedFlow.

Table 5: Training hyperparameters for TamedFlow.

Component	Stage 1	Stage 2
Optimizer	AdamW ($\beta_1 = 0.9, \beta_2 = 0.999$)	
Base learning rate	5×10^{-5}	1×10^{-4}
Weight decay	0.0	
Learning rate schedule	cosine decay with warmup	
Warmup epochs	5	
Max epochs	10	80
Batch size (per GPU)	32	
Global batch size	256	
Gradient clipping	1.0	
Loss terms		
Velocity loss $\mathcal{L}_v$	✓	✓
Alignment loss $\mathcal{L}_{\text{align}}$	$\lambda = 0.25$	–
Rollout loss $\mathcal{L}_{\text{roll}}(K)$	–	$\lambda = 0.8$
Endpoint loss $\mathcal{L}_z$	–	λ_z (10-epoch warmup to 30.0)
Confidence loss $\mathcal{L}_{\text{conf}}$	–	$\lambda = 1.0$
Model details		
Student architecture	DiT-B/2 (12 layers)	
Student parameters	130M	
Teacher architecture	DiT-XL/2 (28 layers)	
Teacher parameters	675M	
AlignNet architecture	3 independent MLPs, $768 \to 1152 \to 1152$	
VAH architecture	3-layer Conv2D residual head, hidden=256	
LTE architecture	3 projections; MLP $1024 \to 1024 \to 256 \to 1$	
LTE parameters	~4M (3% of student)	
Inference hyperparameters		
Draft length K	3 (training: $K \sim \mathcal{U}(1, 5)$)	
Temperature T_z	0.5	
Temperature T_v	0.1	
Mixing weight α	0.7	
Default τ	0.8	
LTE training		
Gradient isolation	–	✓(stop-gradient)
LTE warmup epochs	5	
LTE learning rate	1×10^{-4}	

Appendix B Baseline Comparison Scope

Table 1 compares end-to-end operating points, not isolated routers. The adaptive baselines retain the 675M teacher, whereas TamedFlow combines a 130M draft student with selective teacher rescue; compression, training, solver choice, routing, and rollback therefore all contribute to the measured result. All baseline entries, including SpecDiff, are evaluated in our local implementations under the stated protocol. TamedFlow FLOPs marked with † are backbone-call estimates; end-to-end latency additionally includes VAH, LTE, solver, and rollback overhead.

Table 11 serves a different purpose. It is a quality diagnostic in which adaptive reuse is attached directly to the compressed student, and therefore supports the motivating Combination Gap rather than a latency-matched routing comparison.

Appendix C Higher-Resolution Evaluation

We evaluate spatial scaling on ImageNet at 512×512 with DiT-XL/2, Euler-50, and one A800 GPU. The comparison holds the resolution and protocol fixed and tests whether the primary quality–latency ordering survives the larger latent grid.

Table 6: ImageNet 512×512 evaluation with DiT-XL/2 using fixed-step Euler.

Method	Latency (s) ↓	Speedup ↑	FID ↓	IS ↑
Teacher (DiT-XL/2, 675M)	3.842	1.00×	2.87	235.10
TaylorSeer ($N = 3$)	1.715	2.24×	3.76	226.75
FoCa ($N = 3$)	1.628	2.36×	3.47	228.10
TamedFlow ($\tau = 0.80$)	**1.594**	**2.41×**	**3.39**	**230.85**

Appendix D Multi-Seed Evaluation

We generate 50K ImageNet samples for each of three evaluation seeds using fixed checkpoints. These repeats quantify sampling variability, not independent training-run variability; every method follows the same evaluation protocol.

Table 7: Three-seed ImageNet evaluation (mean ± standard deviation over seeds).

Method	Speedup	FID	IS
Euler-50 Teacher	1.00×	2.36 ± 0.02	237.24 ± 0.48
TaylorSeer ($N = 3$)	2.19×	2.71 ± 0.04	229.13 ± 0.62
FoCa ($N = 3$)	2.21×	2.64 ± 0.03	231.24 ± 0.58
TamedFlow, $\tau = 0.80$	**2.26×**	2.56 ± 0.03	233.89 ± 0.51

Appendix E Efficiency and Transfer Accounting

Deployment footprint. In a controlled A800, batch-size-2, FP32, 50-step latent-only profile with VAE decoding excluded, peak allocated VRAM is 2.570 GB for the teacher, 2.580 GB for FoCa ($N = 3$), 3.066 GB for TaylorSeer ($N = 3, O = 3$), and 3.075 GB for TamedFlow. Co-deploying the student and LTE therefore adds about 0.5 GB relative to the teacher-only profile. For the ImageNet and MS-COCO image experiments, all methods use classifier-free guidance at scale 3.5. The teacher FLOP reference includes the complete conditional and unconditional CFG evaluation, while the CFG-distilled student reference counts one student forward; this convention is used consistently in the reported backbone accounting.

Table 8: Teacher-rescue counterfactual on ImageNet at $K = 3$.

Rescue policy	Speedup	FID
Full TamedFlow, teacher rescue	2.25×	**2.58**
Student rescue (Student+VAH)	–	7.12
Dense student, no routing	5.80×	5.56

Routing accounting. At $\tau = 0.80$, 14 of 22 routing events are accepted and 8 are rejected, giving a 63.6% event acceptance rate. Since each accepted event advances $K = 3$ Student steps, accepted drafts cover $42/50 = 84\%$ of the nominal trajectory; the remaining 8 steps use Teacher rescue. The separately measured component profile records 0.07075 s for accepted-draft buckets (14.9%), 0.06031 s for rejected-draft buckets including rollback (12.7%), 0.34142 s for Teacher rescue (71.9%), and 0.00237 s for LTE scoring (0.5%). These buckets sum to 0.47485 s (up to rounding). They are event-level timing buckets from the profiling run, so dividing a bucket by the number of backbone calls

would incorrectly attribute rollback and routing overhead to an individual forward. The component profile and the headline latency measurement were collected in different runs: the 0.464 s headline is the separately measured ten-run average. We therefore report both traces without using one to reconstruct the other.

Latte video protocol. We retrain the student, AlignNet, VAH, and LTE for the evaluated Latte video-DiT setting while keeping the TamedFlow training and inference configuration consistent with the primary experiments: Stage 1 for 10 epochs, Stage 2 for 80 epochs, $K = 3$ at inference, randomized $K \sim \mathcal{U}(1, 5)$ during rollout training, and the same loss weights and threshold as ImageNet. The original Latte teacher/student block configurations are retained; no width, depth, or temporal-factorization search is performed. Spatial and temporal tokens are treated as one spatiotemporal token sequence for AlignNet matching and LTE mean pooling, so the LTE adds no separate temporal backbone. Evaluation uses FaceForensics clips at 256×256 resolution with 16 frames. We use the same clips and evaluation protocol for all methods, batch size 1, deterministic 50-step sampling, and latency measured end-to-end on 8 A800-80GB GPUs. VAE decoding is excluded from the reported latency. The video table reports FVD and video Inception Score together with the teacher-normalized speedup.

MS-COCO adaptation protocol. The shared 19.9M text-conditioning adapter is trained jointly with the teacher and student for 20 epochs (5 Stage 1 and 15 Stage 2). The implementation uses AdamW with learning rate 10^{-4} for the text adapter, a 0.1 learning-rate multiplier for the backbone parameters, weight decay 0.01, text-dropout probability 0.05, gradient clipping at 1.0, and EMA decay 0.9999. The DiT backbones are frozen except for the text-conditioning parameters; the LTE and VAH are frozen during this adapter-only adaptation. The adapter maps 77-token CLIP ViT-L/14 features into the DiT conditioning width and is shared by teacher and student.

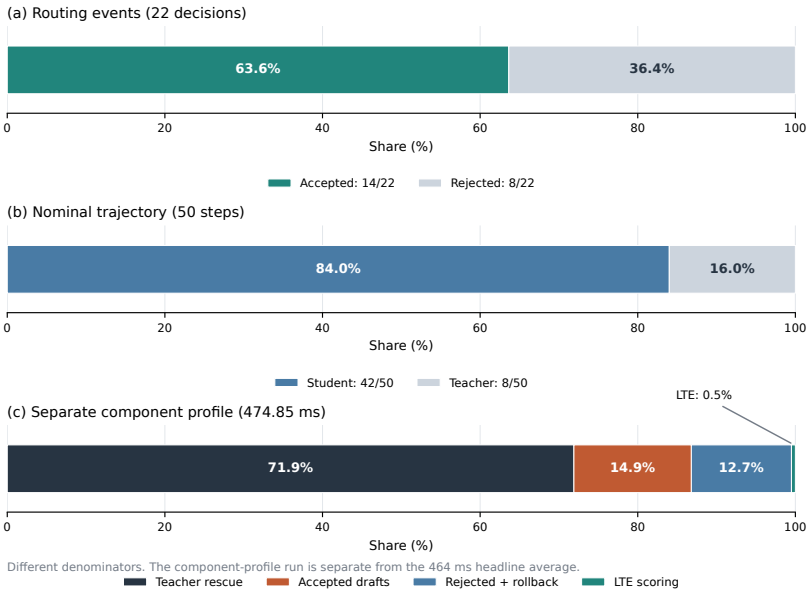


Figure 5: Compact routing and execution accounting at $K = 3$, $\tau = 0.80$. The panels use distinct denominators: (a) 14 accepted and 8 rejected routing events; (b) 42 student-covered and 8 teacher-rescue nominal steps; and (c) the separately collected component profile. The component profile totals 474.85 ms and is not a decomposition of the separate 464 ms headline average.

Appendix F Proof of the K-Step Trajectory Alignment Bound

Let $x_S^{(k)}$ and $x_T^{(k)}$ denote the states of the student and teacher at step k during a K-step sequential Euler rollout, starting from the identical initial state $x_S^{(0)} = x_T^{(0)} = x_t$. We assume a fixed ODE

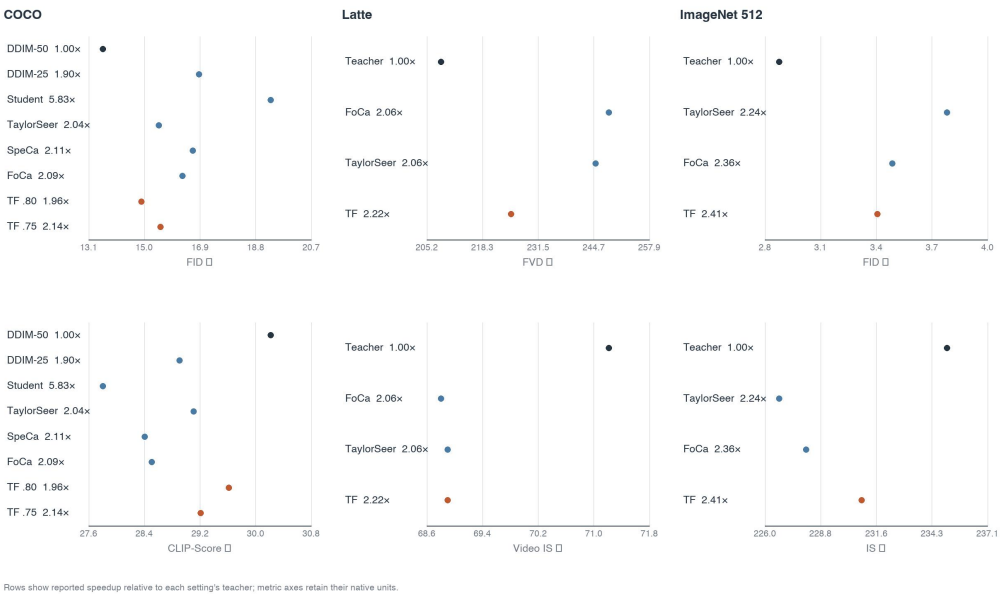


Figure 6: Cross-setting transfer shown as six compact native-unit panels. Columns are COCO, Latte/FaceForensics, and ImageNet 512×512; the top row reports quality metrics and the bottom row reports semantic or recognition scores. Row labels include the reported speedup relative to the teacher for that setting.

solver step size h. With time arguments suppressed at the common $t_k = t + kh$, define the baseline velocity mismatch evaluated strictly on the teacher's trajectory as $\delta_v^{(k)} \triangleq \tilde{v}_S(x_T^{(k)}) - v_T(x_T^{(k)})$, with its supremum bounded as $\|\delta_v\|_\infty \triangleq \max_k \|\delta_v^{(k)}\|_2$. Define the teacher's Lipschitz constant $L_T \triangleq \sup_{x,t} \|J_{v_T}(x,t)\|_2$, and the Jacobian mismatch $\Delta J \triangleq \sup_{x,t} \|J_{\tilde{v}_S}(x,t) - J_{v_T}(x,t)\|_2$.

Proof. **Setup: recursive error accumulation.** In a multi-step rollout, each model feeds its own prediction into the next Euler step:

$$x_S^{(k+1)} = x_S^{(k)} + h\,\tilde{v}_S(x_S^{(k)}), \qquad x_T^{(k+1)} = x_T^{(k)} + h\,v_T(x_T^{(k)}). \tag{F.1}$$

Let $e_k \triangleq x_S^{(k)} - x_T^{(k)}$ denote the spatial deviation at step k, with the initial condition $e_0 = 0$. Subtracting the two updates yields the discrete error dynamics:

$$e_{k+1} = e_k + h\left(\tilde{v}_S(x_S^{(k)}) - v_T(x_T^{(k)})\right). \tag{F.2}$$

Decomposing the velocity difference via integral Taylor expansion. To separate the pointwise baseline error from the accumulated spatial drift, we add and subtract the student's velocity at the teacher state, $\tilde{v}_S(x_T^{(k)})$:

$$\tilde{v}_S(x_S^{(k)}) - v_T(x_T^{(k)}) = \underbrace{\left(\tilde{v}_S(x_T^{(k)}) - v_T(x_T^{(k)})\right)}_{\delta_v^{(k)}} + \underbrace{\left(\tilde{v}_S(x_S^{(k)}) - \tilde{v}_S(x_T^{(k)})\right)}_{\text{Drift induced by } e_k}. \tag{F.3}$$

Since the velocity field $\tilde{v}_S : \mathbb{R}^D \to \mathbb{R}^D$ is vector-valued, the scalar Mean Value Theorem does not apply. We instead write the spatial drift using the first-order Taylor expansion with an integral remainder along the line segment connecting $x_T^{(k)}$ and $x_S^{(k)}$:

$$\tilde{v}_S(x_S^{(k)}) - \tilde{v}_S(x_T^{(k)}) = \left(\int_0^1 J_{\tilde{v}_S}\left(x_T^{(k)} + \tau e_k\right) \mathrm{d}\tau\right) e_k. \tag{F.4}$$

Taking the L_2 norm of the error recurrence, substituting the decomposition and the integral expansion, and applying the triangle inequality yields:

$$\begin{aligned}\|e_{k+1}\|_2 &\le \|e_k\|_2 + h\,\|\delta_v\|_\infty + h\left\|\int_0^1 J_{\tilde{v}_S}\big(x_T^{(k)} + \tau e_k\big)\,\mathrm{d}\tau\right\|_2 \|e_k\|_2 \\ &\le \|e_k\|_2 + h\,\|\delta_v\|_\infty + h\left(\int_0^1 \sup_x \|J_{\tilde{v}_S}(x)\|_2\,\mathrm{d}\tau\right)\|e_k\|_2.\end{aligned} \tag{F.5}$$

Here, $\sup_x \|J_{\tilde{v}_S}(x)\|_2$ is the student's Lipschitz constant L_S. By splitting the student's Jacobian into the teacher term and the mismatch term, we bound it with the matrix triangle inequality:

$$L_S = \sup_{x,t} \|J_{\tilde{v}_S}\|_2 = \sup_{x,t} \|J_{v_T} + (J_{\tilde{v}_S} - J_{v_T})\|_2 \le L_T + \Delta J. \tag{F.6}$$

Since the integral over $[0,1]$ equals one, substituting the Jacobian bound into the preceding inequality gives the recurrence:

$$\|e_{k+1}\|_2 \le \big(1 + h(L_T + \Delta J)\big)\|e_k\|_2 + h\,\|\delta_v\|_\infty. \tag{F.7}$$

Solving the recurrence. Let $C \triangleq L_T + \Delta J$. Unrolling the recurrence with the initial condition $e_0 = 0$ yields a geometric series:

$$\|e_K\|_2 \le h\,\|\delta_v\|_\infty \sum_{i=0}^{K-1}(1 + hC)^i = \frac{\|\delta_v\|_\infty}{C}\big((1 + hC)^K - 1\big). \tag{F.8}$$

To expose the polynomial dependence on K in the few-step regime, we use the Taylor expansion of the exact scalar term $(1 + hC)^K$:

$$(1 + hC)^K = 1 + K(hC) + \frac{K(K-1)}{2}(hC)^2 + \mathcal{O}\big(K^3(hC)^3\big). \tag{F.9}$$

For $C = 0$ the recurrence gives $\|e_K\|_2 \le Kh\|\delta_v\|_\infty$. For $KhC \ll 1$, expansion of the geometric upper bound gives:

$$\|e_K\|_2 \le Kh\|\delta_v\|_\infty + \frac{K(K-1)}{2}h^2C\|\delta_v\|_\infty + \mathcal{O}\big(K^3h^3C^2\|\delta_v\|_\infty\big). \tag{F.10}$$

Interpretation. The exact bound isolates the baseline velocity mismatch and shows how the combined Lipschitz/mismatch constant $C = L_T + \Delta J$ amplifies errors with the rollout horizon. In the few-step expansion, this produces the stated quadratic-in-K sensitivity. Pointwise velocity matching directly supervises the forcing term but does not explicitly constrain the spatial amplification constant; it may nevertheless affect both. The expansion concerns an upper bound, not a measured error growth law. □

Appendix G Scope of the VAH Mechanism

VAH is a residual correction trained through the Stage-2 velocity and trajectory objectives in Eq. equation 5. No separate explicit Jacobian-matching objective is specified by this training recipe. For $v_S^{\text{aligned}} = v_S + \text{VAH}(v_S)$, the chain rule gives $J_{v_S^{\text{aligned}}} = (I + J_{\text{VAH}})J_{v_S}$. This permits changes to local derivatives but does not guarantee reduced Jacobian mismatch. Figure 9 reports velocity mismatch and endpoint drift only. It provides no Jacobian diagnostic: reduction of ΔJ remains a theoretical possibility, not an experimentally established mechanism.

Appendix H Inference Details

Algorithm 1 gives the full routing loop under the forward flow-time convention $t_0 < \cdots < t_N$. The student drafts up to K steps without teacher access, and the LTE scores the draft from the last student forward, the same type of input used during LTE training. Rejection restores the pre-draft state before one teacher rescue step; no endpoint backbone call is added for scoring.

Algorithm 1 TamedFlow Inference

Input: $x \sim \mathcal{N}(0, I)$, y, discrete times $\{t_0, \ldots, t_N\}$, $K = 3$, threshold τ
$i \leftarrow 0$
while $i < N$ **do**
  $m \leftarrow \min(K, N - i)$ ▷ Clip the terminal draft horizon
  $x_{\text{start}} \leftarrow x$
  for $k = 1$ **to** m **do** ▷ Student drafts m steps
    $(v_S, \text{tokens}_k) \leftarrow \mathcal{B}_\theta(x, t_{i+k-1}, y)$
    $v_S \leftarrow v_S + \text{VAH}(v_S)$
    $x \leftarrow x + (t_{i+k} - t_{i+k-1}) \cdot v_S$
  end for
  $c \leftarrow \text{LTE}(\overline{\text{tokens}}_m, e_{t_{i+m-1}}, e_y)$ ▷ Last draft features; no teacher access
  if $c \geq \tau$ **then**
    $i \leftarrow i + m$ ▷ Accept: advance by m, no teacher call
  else
    $x \leftarrow x_{\text{start}}$ ▷ Reject: roll back to pre-draft state
    $v_T \leftarrow \mathcal{T}_\phi(x, t_i, y)$ ▷ Teacher rescue evaluation step
    $x \leftarrow x + (t_{i+1} - t_i) \cdot v_T$
    $i \leftarrow i + 1$
  end if
end while
return x

Appendix I LTE Calibration and Routing Metrics

Table 9 reports held-out routing diagnostics at $\tau = 0.80$ and $K = 3$. The LTE regresses the continuous target $c^\star$ in Eq. equation 7; binary labels are introduced only to summarize thresholded routing errors.

Reference labels and routing errors. A draft event is labeled offline as accept when $c^\star \geq 0.80$ and reject otherwise; the online decision accepts when the prediction $c \geq 0.80$. FAR uses the 4,683 reference-reject events as its denominator, FRR uses the 11,984 reference-accept events, and AUC ranks events by continuous c. These labels measure agreement with the specified teacher-consistency target, not preservation of final sample quality.

Validation protocol. Threshold and score-temperature choices are selected on a calibration split and then frozen. Table 9 is computed on a held-out trajectory-level split, preventing draft events from one trajectory from appearing in both sets. Continuous-target ECE is the event-weighted 10-bin discrepancy between mean c and mean $c^\star$; it measures score agreement rather than a probability of correct generation.

Table 9: LTE classification and calibration metrics on the ImageNet 256×256 validation set at $\tau = 0.80$, $K = 3$. Support counts reference labels obtained by thresholding $c^\star$ at 0.80. FAR and FRR use reference-reject and reference-accept events as their respective denominators. AUC uses continuous predictions; continuous-target ECE uses 10 confidence bins.

Support (Reference)	**Accept**	**Reject**
Count	11,984	4,683
Metric	**Value**	
False Accept Rate (FAR)	0.063	
False Reject Rate (FRR)	0.051	
AUC	**0.94**	
Continuous-target ECE (10 bins)	0.034	

The reference-positive fraction is $11{,}984/16{,}667 = 71.9\%$, whereas the online predicted acceptance rate is 63.6%. These use different labels and event populations, so their difference alone does not identify distribution shift. Held-out diagnostics do not establish error rates along free-running online trajectories. Thresholding predicted confidence separates routed events by construction; these held-out diagnostics do not establish calibration against actual draft error.

Appendix J Additional Ablation Study

Table 10 extends the main ablation with randomized-horizon variants. Unless stated otherwise, rows use ImageNet 256×256, $K = 3$, and $\tau = 0.80$.

Table 10: Protocol-consistent extended ablation on ImageNet 256×256. Inference uses $K = 3$, $\tau = 0.80$; the randomized horizons refer to training.

Variant	Accept Rate % ↑	Speedup ↑	FID ↓	IS ↑
Full TamedFlow (default)	63.6	2.25×	**2.58**	**233.43**
Module and loss ablations				
w/o AlignNet	46.2	1.87×	2.95	211.34
w/o VAH	39.3	1.63×	3.23	205.71
w/o $\mathcal{L}_{\text{roll}}$	54.2	2.02×	2.71	230.15
w/o $\mathcal{L}_z$ warmup	46.2	1.81×	3.29	203.50
w/o $\mathcal{L}_z$	28.1	1.45×	4.15	198.80
K **strategy ablations**				
Random $K \sim \mathcal{U}(1,3)$	75.0	2.14×	2.53	233.22
Random $K \sim \mathcal{U}(1,5)$ (default)	63.6	2.25×	2.58	233.43
Pure Student (no LTE, no routing)	–	5.80×	5.56	195.07

Table 11: Same-student validation under the common Euler-50 protocol. Only FID is reported because these direct adaptive baselines were evaluated as quality diagnostics, not as the headline latency comparison.

Strategy	FID	Strategy	FID
FoCa	10.40	TaylorSeer	11.92
SpeCa	13.18	Dense Student	5.56

The extended rows reinforce the main ablation: representation alignment, residual correction, rollout loss, and endpoint loss each improve the usable routed operating point. Randomizing training over $K \sim \mathcal{U}(1,3)$ yields a nearby but distinct point (FID 2.53 at $2.14\times$), whereas the wider default $\mathcal{U}(1,5)$ reaches FID 2.58 at $2.25\times$.

At fixed $K = 3$, the 75.0% point corresponds to 15 accepted and 5 rejected events (60 student and 5 teacher calls under full-length drafts), whereas the 63.6% point corresponds to 14 accepted and 8 rejected events (66 student and 8 teacher calls). Thus the call-count model predicts lower backbone work for the higher-acceptance point. Its measured $2.14\times$ versus $2.25\times$ ordering was obtained from separate checkpoint/runtime measurements, and the available component trace does not provide a matched per-call timing decomposition; we therefore do not interpret this ordering as a monotonic acceptance–latency law.

Under the common Euler-50 diagnostic, directly attaching FoCa, TaylorSeer, or SpeCa to the compressed student degrades FID relative to dense student execution. This documents the Combination Gap for these implementations; it does not test whether the same methods remain fragile after trajectory training.

For the DPM-Solver++(2M) diagnostic, the multistep history is associated with the current accepted state. When a student draft is rejected, the tentative student states and derivative history are discarded together; the sampler rolls back, applies one teacher rescue update, and starts a fresh solver history at the rescued state. No student derivative is mixed with a teacher derivative across a rejection boundary,

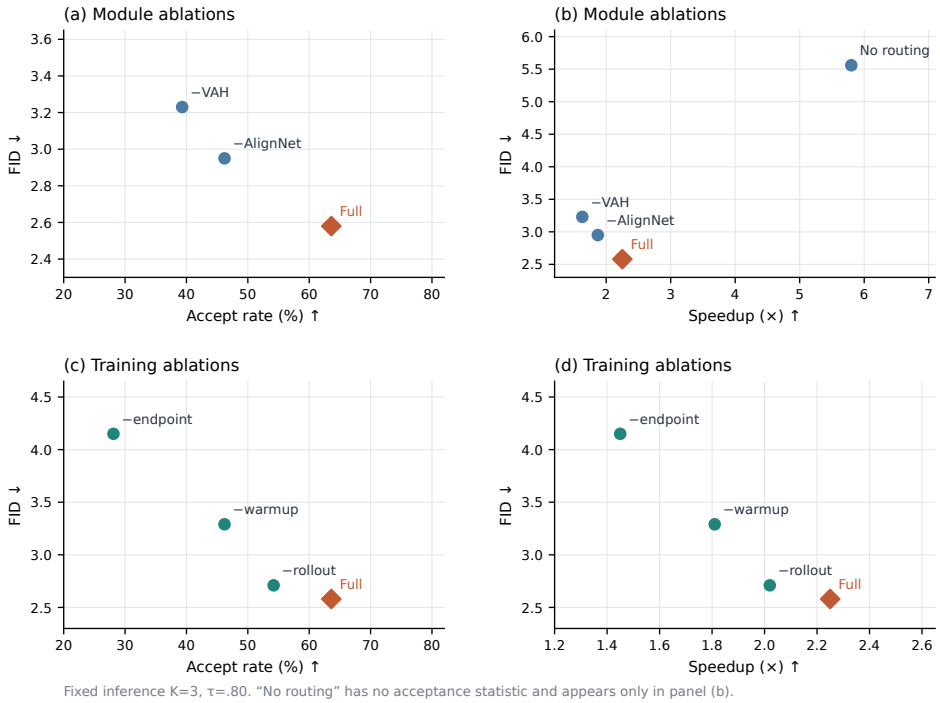


Figure 7: Module and training ablations from Tables 2 and 10. The panels relate FID to acceptance or speedup at $K = 3$, $\tau = 0.80$. The no-routing student has no acceptance statistic and is omitted from the acceptance panels. Points are reported operating points; no fitted trend or uncertainty interval is implied.

Table 12: Solver robustness on ImageNet 256×256. DPM-Solver++(2M) is evaluated as a handoff-safe higher-order solver diagnostic; the formal bound remains Euler-specific.

Solver	Method	Latency	Speedup	Accept	FID	IS
Euler	Teacher	1.044	1.00×	–	2.35	237.62
Euler	TamedFlow	0.464	2.25×	63.6%	2.58	233.43
DPM-Solver++(2M)	Teacher	1.089	1.00×	–	2.25	244.12
DPM-Solver++(2M)	TamedFlow	0.455	2.39×	75.0%	2.44	237.55

and the first post-handoff update uses the solver's order-one startup rule before the two-step history is rebuilt. Thus Table 12 tests end-to-end robustness under an explicit reset policy; Proposition 3.1 continues to cover only the Euler analysis.

Figure 9 reports velocity and endpoint errors, not Jacobian alignment. Removing AlignNet or VAH changes the measured velocity or endpoint mismatch, while removing rollout supervision increases short-horizon drift.

Figure 10 visualizes the impact of draft horizon K and routing threshold τ on generation quality and inference efficiency. The four marginal panels show FID and speedup versus τ (top) and K (bottom). These one-dimensional sweeps motivate the default operating point but do not establish a globally optimal Pareto frontier.

Appendix K Limitations

While TamedFlow provides empirical gains across several settings, our findings are subject to several key limitations:

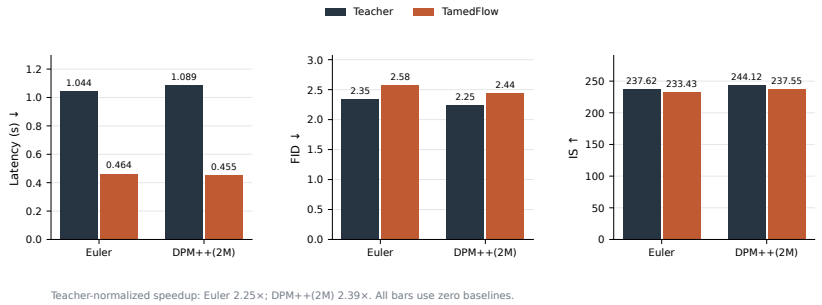


Figure 8: Paired teacher and TamedFlow measurements under Euler and DPM-Solver++(2M), reproducing Table 12. Speedups are relative to the corresponding solver teacher. The DPM++ result uses the stated history-reset policy; the theoretical bound remains Euler-specific.

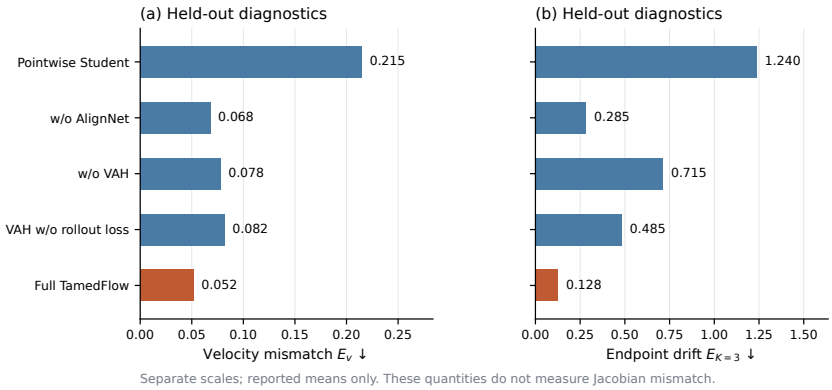


Figure 9: Held-out velocity mismatch and three-step endpoint drift by model variant. The panels replace the redundant numerical table and use separate zero-based axes. These means do not measure Jacobian mismatch or establish that training minimizes the theoretical bound constants; no uncertainty estimates are available.

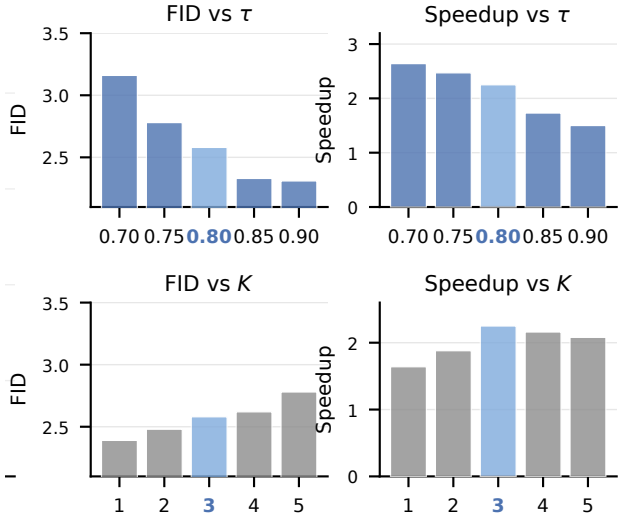


Figure 10: Impact of draft length K and routing threshold τ on generation quality and inference efficiency. Columns: FID (left) and speedup (right). Rows: threshold τ (top) and draft length K (bottom). The sweeps motivate the default $K = 3$, $\tau = 0.80$ configuration.

Theoretical scope of rollout analysis. The discrete Grönwall bound in Proposition 3.1 characterizes error accumulation under sequential Euler drafting to motivate multi-step trajectory supervision. It does not constitute a universal failure law for internal feature caching or forecasting methods, nor does it provide formal convergence or stability guarantees for higher-order ODE solvers such as DPM-Solver++.

Approximate nature of trajectory routing. The Learnable Trajectory Estimator (LTE) operates as an empirical regression model trained on offline consistency targets rather than an exact, distribution-preserving verification mechanism. Rejection decisions are heuristic; false rejections incur latency penalties through redundant teacher evaluations, whereas false acceptances may allow minor perceptual artifacts to propagate into the generated sample.

Deployment and infrastructure overhead. TamedFlow relies on a multi-stage offline training curriculum and maintains the teacher, student, VAH, and LTE models co-resident in GPU memory during inference (~ 0.5 GB VRAM overhead in our FP32 profile). As a result, it does not fully eliminate the teacher model from deployment. Furthermore, our reported speedups reflect batch-size-one latent generation on single GPUs; characterization under large-batch throughput, concurrent serving pipelines, and extreme memory-constrained edge environments remains an avenue for future work.

Appendix L LLM Usage Statement

Large language models were used for language editing and consistency checks across the manuscript. Experimental results are author-provided; the authors are responsible for checking the final equations, measurements, citations, and implementation details.