PARAMETRIC TRAJECTORY DISTILLATION FOR FEW-STEP VIDEO GENERATION

Anonymous authors
Paper under double-blind review



Figure 1: **Natural motion and seed diversity in four steps (MiniMax-H3).** Like the teacher, PTD carves deep turns with seeds that differ in framing and background, whereas the two LightX2V Turbo (DMD) seeds nearly repeat framing and pose. Over 100 prompts, PTD raises GPT-6 Astra diversity from 1.55 to 2.81 (Section 4.2). Each row: one seed at five matching timestamps. Teacher: 50 steps; students: four. Prompt: Appendix F.1.

ABSTRACT

Video diffusion and flow models require many sequential evaluations, making generation computationally expensive. Few-step distillation reduces this cost but poses a capacity allocation problem: a student must match the teacher's iterative generation with far less sequential computation. Existing trajectory methods ask the student to reproduce teacher transitions that are highly curved at high noise, which can exceed its capacity and degrade fine detail. We introduce Parametric Trajectory Distillation (PTD), which lets the student parameterize teacher trajectory segments as polynomials and learn from teacher guidance along its own predicted path. PTD is designed to let the learned curvature adapt to the backbone's predictive capacity, preserving motion and diversity. The curvature head is used only in training; inference keeps the original backbone architecture. On Wan2.1-14B, four-step PTD sets a new state of the art for trajectory distillation, significantly improving dynamic quality and naturalness over PDD, the best-performing trajectory-only method on this model, under the same training setting. On the 33B audio-video MiniMax-H3, LoRA-trained PTD significantly improves diversity and naturalness over the state-of-the-art LightX2V Turbo. Blinded human votes give PTD 55.1% and 63.4% preference shares against PDD and LightX2V Turbo. Anonymous project page: ptd-website.github.io.

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.

1 INTRODUCTION

Video diffusion and flow models generate samples through many sequential network evaluations (Team Wan, 2025; Lipman et al., 2023; Liu et al., 2023). Few-step distillation compresses this iterative process into a handful of student evaluations, often retaining the teacher's backbone. The student may therefore have the same parameter count but far less sequential computation with which to reproduce the teacher's generation process. This makes distillation a problem of capacity allocation: supervision must help the student use its limited computation to approximate a much longer generation process at comparable sample quality.

Existing approaches address this challenge through trajectory supervision or distribution matching. Trajectory objectives (Song et al., 2023; Lu & Song, 2025) train the student to model the teacher's full noise-to-sample trajectory, preserving motion and diversity. This objective is mode-covering: when a few-step student lacks the capacity to represent complex teacher transitions, regression pulls its predictions toward an average of the targets, blurring fine detail (Zheng et al., 2026). Recently, Parallel Decoding Distillation (PDD) achieved strong visual detail with trajectory supervision alone by regressing parallel interval-velocity outputs to teacher Runge–Kutta targets (Anonymous, 2026). Distribution Matching Distillation (DMD) avoids this averaging: its mode-seeking objective evaluates the teacher only on the student's own samples, recovering sharp detail at the cost of substantially reduced sample diversity (Yin et al., 2024b; Zheng et al., 2026). Hybrid methods such as rCM and AnyFlow combine the two paradigms (Zheng et al., 2026; Gu et al., 2026), but the added objectives increase training cost and complexity. Rather than stacking objectives, is there a more elegant way to combine the strengths of both paradigms?

We introduce *Parametric Trajectory Distillation* (PTD) to align trajectory learning with the student's predictive capacity. A single student evaluation predicts a polynomial approximation of a teacher trajectory segment, and teacher velocities are queried along this predicted path (Figure 2). Like distribution matching, which evaluates the teacher only on the student's own samples, PTD thus supervises the student on the path it currently predicts, while keeping the noise-to-sample correspondence of trajectory supervision. This design is built to retain the teacher's motion and diversity without demanding trajectory detail that the student's backbone cannot reliably predict. The curve serves as a training scaffold: its mean velocity directly supplies the endpoint update, while the curvature head is discarded at inference, preserving the original inference architecture. Both predictions are trained with a single velocity-regression loss, without computing Jacobian-vector products or training an auxiliary critic. These properties make PTD a simple recipe for scaling trajectory distillation to large backbones, including through parameter-efficient LoRA fine-tuning.

We distill MiniMax-H3 (MiniMax, 2026), a 33B joint audio-video model, into a four-step generator using LoRA and trajectory supervision alone. Against LightX2V Turbo (ModelTC, 2026), a state-of-the-art four-step DMD method on the H3 Acceleration Arena (multimodalart, 2026), PTD significantly improves diversity (2.81 vs. 1.55) and naturalness as measured by GPT-6 Astra, while matching its dynamic and static quality (Figures 1 and 4). We also apply PTD with full fine-tuning to Wan2.1-14B (Team Wan, 2025), a 14B text-to-video model. Under the same training setting, it outperforms PDD (Anonymous, 2026), the best-performing trajectory-only method reported on this model, significantly improving dynamic quality and naturalness under the same judge and achieving a higher overall VBench score (Huang et al., 2024). Blinded human votes give PTD 63.4% and 55.1% preference shares against LightX2V Turbo and PDD.

Our contributions are: (i) a parametric trajectory distillation formulation that queries teacher velocities along student-predicted polynomial paths and retains only the predicted mean velocity at inference; (ii) analyses showing that teacher guidance along the student's own path preserves fine detail and that the student learns a shrunken, learnable version of the teacher's curvature; and (iii) state-of-the-art four-step video distillation with trajectory supervision alone, improving over LightX2V Turbo on the 33B MiniMax-H3 and surpassing PDD on Wan2.1-14B.

2 RELATED WORK

Endpoint and flow-map learning. Consistency models learn flow maps (Song et al., 2023; Kim et al., 2024), whose semigroup property underlies progressive distillation and shortcut models (Salimans & Ho, 2022; Frans et al., 2025; Boffi et al., 2025b). MeanFlow and its follow-ups learn

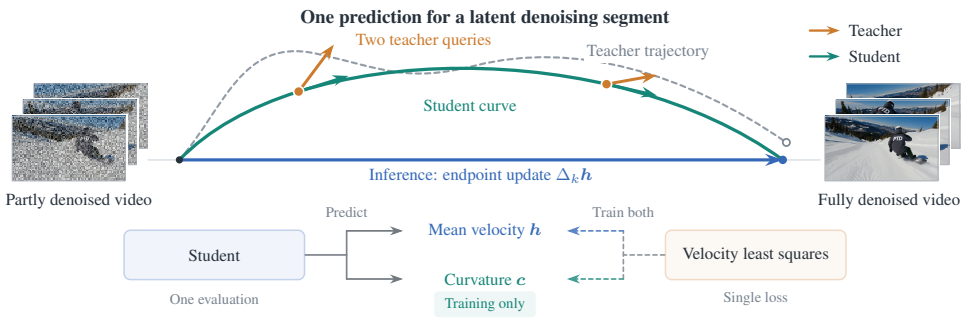

Figure 2: **Parametric Trajectory Distillation.** One student evaluation predicts mean velocity $\boldsymbol{h}$ and curve coefficients $\boldsymbol{c}$ for a latent denoising segment. Teacher velocities queried on the student curve train both outputs with one least-squares loss. Inference uses only $\boldsymbol{h}$ for the endpoint update; the curvature head is removed. Illustrative video thumbnails depict a segment ending at the data endpoint; the same construction applies between intermediate noise levels. Arrows show step-scaled velocities; two queries are the default.

interval-average velocity (Geng et al., 2025; 2026; Guo et al., 2025). PTD's $\boldsymbol{h}$ is this mean velocity, fit to teacher velocities rather than to self-consistency targets.

Explicit trajectories and local supervision. These methods differ in where the teacher is queried. DSNO learns continuous trajectories from teacher-generated states (Zheng et al., 2023); PDD, the best-performing trajectory-only method on Wan2.1-14B, regresses fusible interval-velocity outputs to teacher Runge–Kutta targets (Anonymous, 2026). Flow Map Matching and Physics Informed Distillation query the teacher at student-predicted states (Boffi et al., 2025a; Tee et al., 2024); π-Flow imitates teacher velocities along rollouts of a compact policy integrated at inference (Chen et al., 2026). PTD queries the teacher along its own predicted curve, takes velocities from analytic basis derivatives instead of differentiating the student in time, and drops the curvature head at inference.

Distribution matching for video. DMD and DMD2 match generated distributions through score-based supervision (Yin et al., 2024b;a); for video, rCM regularizes consistency with score supervision (Zheng et al., 2026), and AnyFlow adds on-policy distribution refinement (Gu et al., 2026). PTD uses trajectory supervision alone, without any distribution-level loss, and still significantly improves Astra diversity and naturalness over a state-of-the-art DMD student (Section 4.2). Appendix E expands these comparisons.

3 Parametric Trajectory Distillation

PTD predicts a denoising segment in one student evaluation and queries the teacher along that predicted path (Figure 2). A single velocity-regression objective trains both its displacement and interior shape; inference retains only the endpoint update.

Setup. Let $u_{\mathrm{T}}(\boldsymbol{x}, t \mid y)$ be a pretrained flow model conditioned on a text prompt y (Lipman et al., 2023; Liu et al., 2023). The teacher generates samples by integrating $\mathrm{d}\boldsymbol{x}/\mathrm{d}t = u_{\mathrm{T}}(\boldsymbol{x}, t \mid y)$ from noise at $t = 1$ to data at $t = 0$. We distill this process onto a fixed decreasing grid $t_0 > \cdots > t_N$, with $\Delta_k = t_{k+1} - t_k < 0$. We refer to each interval, which one student evaluation traverses, as a *hop*. For hop k, starting at state $\boldsymbol{x}_k$ at time t_k, define normalized progress $\tau \in [0, 1]$ and physical time $s(\tau) = t_k + \tau\Delta_k$. We suppress the conditioning where clear.

3.1 One prediction for a trajectory segment

A single forward pass of the student with parameters θ at $(\boldsymbol{x}_k, t_k)$ predicts a mean-velocity vector $\boldsymbol{h}_\theta \in \mathbb{R}^D$ and K curvature coefficient vectors $\boldsymbol{c}_\theta = (\boldsymbol{c}_{\theta,0}, \ldots, \boldsymbol{c}_{\theta,K-1})$ with $\boldsymbol{c}_{\theta,j} \in \mathbb{R}^D$, where D is

the latent dimension. These outputs define the entire segment:

$$\boldsymbol{x}_{\theta,k}(\tau) = \boldsymbol{x}_k + \Delta_k \left[\tau \boldsymbol{h}_\theta + \sum_{j=0}^{K-1} b_j(\tau) \boldsymbol{c}_{\theta,j} \right], \qquad b_j(\tau) = \tau(\tau-1) P_j(2\tau-1). \tag{1}$$

The *curvature basis functions* b_j are built from degree-j Legendre polynomials P_j and vanish at both endpoints: $b_j(0) = b_j(1) = 0$. Thus the path connects $\boldsymbol{x}_k$ to $\boldsymbol{x}_k + \Delta_k \boldsymbol{h}_\theta$ for any $\boldsymbol{c}_\theta$. The curvature output $\boldsymbol{c}_\theta$ controls only the interior shape, with $\boldsymbol{c}_\theta = 0$ recovering the straight chord.

Mean velocity and local velocity. Advancing a hop in one step requires its mean velocity, while teacher queries provide local velocities. We relate these quantities by differentiating the curve with respect to physical time, holding the network outputs fixed:

$$\boldsymbol{v}_{\theta,k}(\tau) = \frac{\mathrm{d}\boldsymbol{x}_{\theta,k}}{\mathrm{d}s} = \boldsymbol{h}_\theta + \sum_{j=0}^{K-1} d_j(\tau) \boldsymbol{c}_{\theta,j}, \qquad d_j(\tau) = \frac{\mathrm{d}b_j(\tau)}{\mathrm{d}\tau}. \tag{2}$$

Only the known basis functions are differentiated, so additional curve positions and velocities require no further student forward pass or network Jacobian-vector product. The endpoint constraints imply $\int_0^1 d_j(\tau)\,\mathrm{d}\tau = 0$, and hence

$$\int_0^1 \boldsymbol{v}_{\theta,k}(\tau)\,\mathrm{d}\tau = \boldsymbol{h}_\theta, \qquad \boldsymbol{x}_{\theta,k}(1) = \boldsymbol{x}_k + \Delta_k \boldsymbol{h}_\theta. \tag{3}$$

Thus $\boldsymbol{h}_\theta$ is exactly the mean velocity of the represented student curve, for any network weights.

3.2 A SINGLE LOSS ON THE STUDENT CURVE

To train the segment, we sample m positions $\tau_i \in (0,1)$ and query the teacher at the corresponding points on the current student curve:

$$\boldsymbol{x}_i = \mathrm{sg}[\boldsymbol{x}_{\theta,k}(\tau_i)], \qquad s_i = s(\tau_i), \qquad \boldsymbol{u}_i = \mathrm{sg}[u_{\mathrm{T}}(\boldsymbol{x}_i, s_i \mid y)], \tag{4}$$

where sg denotes stop-gradient. We fit the predicted velocity to these teacher velocities with one least-squares loss:

$$\mathcal{L}_{\mathrm{PTD}} = \sum_{i=1}^{m} \omega_i \left\| \boldsymbol{h}_\theta + \sum_{j=0}^{K-1} d_j(\tau_i) \boldsymbol{c}_{\theta,j} - \boldsymbol{u}_i \right\|^2, \qquad \omega_i > 0, \quad \sum_i \omega_i = 1. \tag{5}$$

During each update, gradients flow through both outputs in the predicted velocity; the teacher targets and their query states are detached. The queries are recomputed from the student's predictions on subsequent updates. Equation 5 estimates the Lagrangian map distillation loss (Boffi et al., 2025a;b) with grid start times and polynomial maps; Appendix A.4 relates it to TVM, π-Flow, and PDD.

Joint supervision of displacement and shape. Each local velocity residual supervises $\boldsymbol{h}_\theta$ directly and the curve coefficients through the weights $d_j(\tau_i)$. For the uniformly integrated objective, the zero-integral property of d_j makes the expected output gradient train $\boldsymbol{h}_\theta$ toward the teacher velocity averaged along the current student curve. The same residual trains $\boldsymbol{c}_\theta$ to capture the teacher's velocity variation along that path. No separate mean-velocity or curvature target is required. Appendix A.1 derives these output gradients and distinguishes the mean along the student curve from the mean along the true teacher trajectory. Under a Lipschitz condition on the teacher field, the integrated residual bounds the endpoint error; exact agreement throughout the interval recovers the teacher trajectory (Appendix A.3).

Why query the student curve? A straightforward way to supervise $\boldsymbol{h}_\theta$ is to integrate the teacher ODE densely over a hop and regress to the resulting displacement divided by Δ_k. Each such target costs many sequential teacher evaluations and is hardest to learn on high-noise hops, where the teacher trajectory curves most strongly: on H3 teacher trajectories, two curve terms capture only 46% of the within-hop velocity variation near the noise end, versus 90% on the final hop (Figure 8).

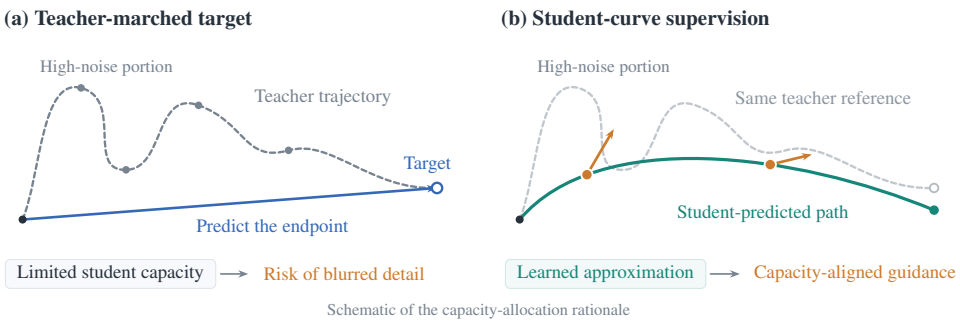

Figure 3: **Where should the student receive teacher supervision? (a)** Dense teacher integration constructs an endpoint target at the cost of many sequential evaluations; on strongly curved, high-noise hops, a limited-capacity student cannot learn this transition well, and fine detail softens. **(b)** PTD queries teacher velocities on the student's current approximation, jointly training its displacement and shape through local guidance. Section 4.4 compares these supervision choices.

Marching every hop softens fine detail (Section 4.4). PTD instead queries teacher velocities along the curve predicted by the student (Figure 3); these local targets avoid dense integration and co-evolve with the predicted coefficients, so supervision always lies on the path the student currently represents. The final hop is much smoother, so H3 marches only that hop; we empirically found that this improves detail at about 20% more training compute (Section 4.2; Appendix B.3), while Wan uses student-curve queries on every hop.

Teacher-query budget. The number m of teacher velocity queries per hop is independent of K, the number of coefficient vectors used to parameterize curvature. At generic query positions, every coefficient participates in the same residual, so each query jointly supervises the full velocity profile (Appendix A.1). Our default draws one point uniformly from each half of the interval ($m = 2$), with equal weights; this is an unbiased estimator of the uniformly integrated squared residual. Additional queries reuse the same student outputs and objective, increasing teacher cost without changing inference. On stored H3 teacher paths, the two-query estimation error is only about one eighth of the student's prediction-error energy (Section 4.4).

3.3 Training and Inference

Training on student rollouts. Starting from sampled noise, we train successive hops on states produced by the student. At each hop, one forward pass predicts the curve, local teacher queries define the loss in equation 5, and the predicted endpoint is detached and carried to the next hop. After the final hop, the rollout restarts from noise. This autoregressive procedure trains the student on its own preceding predictions, using prompts and noise without training videos or precomputed trajectories. Both implementations route gradients within an exact decomposition of equation 5 (Appendix A.2). Appendix B gives the update schedule.

Inference. The endpoint identity in equation 3 gives the sampling rule

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \Delta_k \boldsymbol{h}_\theta(\boldsymbol{x}_k, t_k \mid y), \qquad k = 0, \ldots, N-1. \tag{6}$$

The curvature head is discarded: its role is to shape the supervision used to learn $\boldsymbol{h}_\theta$. Sampling takes N student evaluations, with no teacher queries or interior-curve evaluation. Each deployed hop retains the original backbone topology and output dimensions (Appendix B.4).

4 Experiments

We test whether PTD can improve four-step video quality through trajectory supervision alone. We compare PTD with LightX2V Turbo on MiniMax-H3 (Section 4.2) and with PDD on Wan2.1-14B (Section 4.3). Ablations then examine how supervision and curve parameterization affect learning within the student's limited capacity (Section 4.4).

4.1 BASELINES AND EVALUATION PROTOCOL

Baselines. PDD (Anonymous, 2026) releases neither a training recipe nor weights for MiniMax-H3, so on H3 we compare with LightX2V Turbo (ModelTC, 2026), a state-of-the-art open-source four-step DMD accelerator, using its released v1.1 LoRA and sampler settings. This cross-paradigm comparison tests whether trajectory supervision can retain a DMD student's static quality while producing more varied, natural-looking videos. On Wan2.1-14B, PDD is the best-performing trajectory-only method, so we reproduce it under the same training setting as PTD for a controlled within-paradigm comparison.

Evaluation protocol. Following prior work that treats human judgment as the primary signal (Polyak et al., 2024; Lin et al., 2025), our primary evidence is blinded human votes and GPT-6 Astra judgments, since advanced multimodal LLM judges agree with human preference as well as or better than specialized video metrics (Liu et al., 2025). On Wan2.1-14B, we report VBench (Huang et al., 2024) as a reference: its scores saturate on strong models (Zheng et al., 2025; He et al., 2024; Inbasekar et al., 2026), and its rankings can diverge from human preference (Fan et al., 2026; Han et al., 2025). Astra uses 100 prompts for both comparisons, fixed before video generation: 50 randomly sampled held-out ViMix live-action captions and 50 Claude-generated action prompts; Wan receives only their video descriptions, and each model generates three videos per prompt with shared seeds. Astra, from sampled frames, and human raters, from full playback, compare a prompt's videos as two rows of three, one row per model, with model identities hidden and row placement balanced. Astra scores each model on four 1–4 scales: *dynamic quality* (motion realism, not motion magnitude), *static quality* (per-frame detail and structure, excluding temporal consistency), *naturalness* (color, materials, lighting, and visual rendition), and *diversity* (variation across seeds that respects the prompt). Raters choose the preferred row or a tie, considering visual quality, motion, naturalness, and diversity. Appendices C and G give the full protocol and rubric.

4.2 MINIMAX-H3: COMPARISON WITH LIGHTX2V TURBO

Setup. We distill MiniMax-H3 (MiniMax, 2026), a 33B joint audio-video model, using prompts and noise, without training videos or a distribution-matching objective. Our training prompts mix ViMix (Yang et al., 2025), OpenVid (Nan et al., 2025), and VidProM (Wang & Yang, 2024), rewritten into H3's official prompt format using its released rewriter. We train rank-128 LoRA adapters per hop for 150 updates at batch size 512. The first three hops use $K{=}8$ Legendre terms and two stratified teacher queries on the student curve; the final hop instead regresses to an eight-step teacher-marched endpoint (Appendix B.3). The final hop is far smoother than the noise-end hop (Section 3.2), and we empirically found that marching only this hop improves visual detail over two queries on every hop at about 20% more training compute. Inference discards the curvature outputs and uses four network evaluations. Appendix B specifies the model-specific recipes. Both students generate 124-frame videos at $768{\times}1344$ resolution.

Results. PTD significantly improves Astra diversity from 1.55 to 2.81 and naturalness from 2.98 to 3.09 while matching LightX2V Turbo in dynamic and static quality (Figure 4; Appendix Table 4). Both gains hold on the live-action and action prompt groups (Appendix C). Blinded raters give PTD a 63.4% preference share against LightX2V Turbo. Figure 1 illustrates natural carving motion and variation across seeds in snowboarding.

4.3 WAN2.1-14B: COMPARISON WITH PDD

Setup. We fully fine-tune Wan2.1-T2V-14B (Team Wan, 2025) on randomly sampled ViMix prompts, using two stratified student-curve queries on all four hops. We reproduce PDD strictly following its reported training hyperparameters, with the same backbone, frozen teacher and guidance, ViMix training prompts, global batch size 256, and constant learning rate 10^{-5} as PTD. Both methods generate 81-frame videos at $480{\times}832$ resolution.

Results. Under this shared setting, PTD outperforms PDD: it significantly improves Astra dynamic quality (2.43 vs. 2.16) and naturalness (2.82 vs. 2.60) while matching static quality (Figure 4; Appendix Table 4), and it receives a 55.1% human preference share. After only 64 updates, PTD

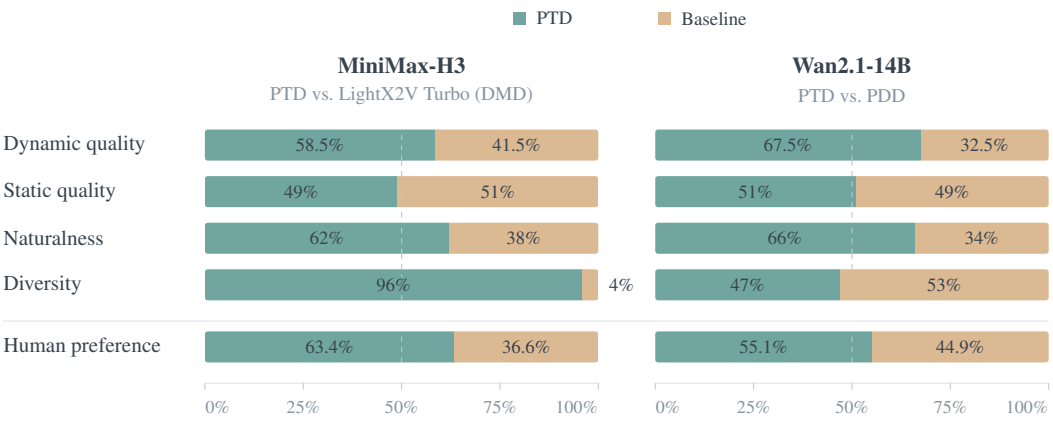

Figure 4: **PTD improves naturalness over both baselines and receives majority human preference shares.** The first four rows show Astra-derived outcomes over 100 prompts, with ties split equally; dashed lines mark 50%. Outcomes use scores averaged over four calls per prompt, with differences below 0.25 in magnitude counted as ties. Human preference is a separate holistic vote (Appendix C). Both evaluations use H3 PTD at 150 updates and Wan PTD at 64 versus PDD at 96. Appendix Table 4 reports the automated scores.

also exceeds the overall VBench score of all three evaluated checkpoints of our PDD reproduction, at 64, 96, and 256 updates (Table 1; Appendix Table 6). We attribute the gap between our PDD reproduction and PDD's published VBench scores to differences in training data and evaluation prompts, together with evaluation noise.

Table 1: **VBench and diversity on Wan2.1-14B.** Under the same training setting, PTD at 64 updates outperforms the best checkpoint of our PDD reproduction (96 updates) in overall, quality, and semantic VBench scores; bold compares these two rows. Gray scores are published reference results from Anonymous (2026).

Method	NFE	VBench ↑			Diversity ↑	
		Overall	Quality	Semantic	V-JEPA 2	VideoMAE
Published reference results						
UniPC* (Teacher)	50×2	83.90	84.56	81.24	0.1263	0.0250
rCM†	4	84.92	85.43	82.88	—	—
AnyFlow*	4	84.95	85.70	81.92	0.0786	0.0130
DMD2** (FastGen)	4	84.40	85.16	81.34	0.0568	0.0095
PDD^{short} (Midpoint)	4	84.92	85.71	81.77	0.0791	0.0125
PDD^{long} (Midpoint)	4	84.69	85.69	80.71	0.0846	0.0126
Our best evaluated checkpoints						
PDD @96 (reproduction)	4	83.67	85.24	77.40	**0.1340**	**0.0254**
PTD @64 (ours)	4	**84.57**	**85.44**	**81.10**	0.1220	0.0180

*Official checkpoints re-evaluated by the PDD authors; **their FastGen implementation. †rCM uses proprietary prompts; other published rows use Self Forcing prompts. PDD short/long denotes 200/3,000 training iterations. NFE (number of function evaluations) includes guidance passes. Published rows use different training budgets and protocols. Diversity is cosine distance in V-JEPA 2 (Assran et al., 2025) and VideoMAE V2 (Wang et al., 2023) feature space; the teacher uses the UniPC solver (Zhao et al., 2023); rCM diversity is not reported in the cited comparison.

Learning within-hop trajectory shape. On eight VBench prompts with a shared 128-slot teacher reference, we measure how well each student learns the teacher's within-hop bend (Figure 5). On the high-noise first hop, PTD has a larger bend magnitude ratio r and a higher direction cosine than PDD at every shared update count from 32 to 224; at 64 updates, r is 0.081 versus 0.024, and the teacher-aligned projection is 0.014 versus 0.003. Final-hop projections remain close. PTD's gain is

largest on this high-noise hop, where trajectory curvature is high and Section 3.2 motivates querying the student curve rather than regressing to teacher-integrated targets. Appendix D.2 gives all-hop results and metric definitions.



Figure 5: **Learning teacher-aligned curvature on Wan2.1-14B.** Bend projection normalized by teacher bend energy (eight prompts). PTD learns a larger aligned component on the high-noise first hop; final-hop projections remain close. Vertical dashed lines mark 64 updates, the evaluated PTD checkpoint.

4.4 Ablations



Figure 6: **Student-curve supervision preserves local detail.** Three Wan2.1-1.3B students after 40 training updates, all using four-step inference, shown at the same two timestamps per prompt. Teacher marching on every hop softens the dog's coat and the painter's hand and brush; both student-curve variants retain distinct detail. Step and query counts refer to teacher supervision per training hop.

Student-curve queries versus teacher marching. We compare three supervision schemes on Wan2.1-1.3B: an eight-step teacher-marched endpoint target on every hop, eight teacher queries on the student curve, and our two-query default. All three students are trained for 40 updates and use four evaluations at inference. The marching baseline integrates the teacher from each carried hop input and trains $\boldsymbol{h}_\theta$ to predict the resulting mean velocity; the student-curve variants jointly train $\boldsymbol{h}_\theta$ and the curve coefficients with the loss in equation 5. Marching on every hop visibly softens local detail around the dog's coat and legs and the painter's hand and brush, whereas both student-curve variants retain distinct structure (Figure 6). As Figure 3 anticipates, a more accurate teacher target is harder for a few-step student to learn, especially on high-noise hops, where teacher trajectories bend most (Section 3.2). Student-curve queries give learnable guidance on these curved hops; on the smoother final hop, we empirically found that marching improves H3's visual detail (Section 4.2).

Accuracy of two-query target estimation. Our default uses two stratified queries per student-curve-supervised hop. On stored H3 teacher trajectories, we compare the average of one velocity

query per half-hop with the endpoint-derived mean. For hops 0–2, the two-query quadrature MSE is only 12.6%, 13.6%, and 13.3% of the student's prediction-error energy (Figure 7), so two queries already give an accurate target. A third query lowers these ratios to 6.6%, 6.1%, and 5.8% at 50% more teacher queries per hop (Appendix D.1).

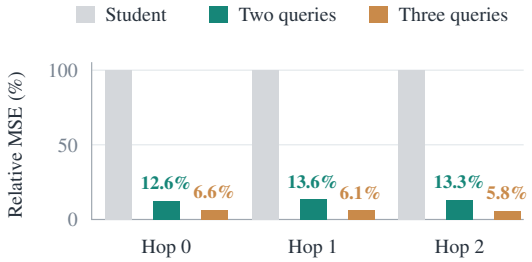

Figure 7: **Two queries give a small target-estimation error.** On video hops 0–2, two-query quadrature MSE is about one eighth of student MSE; three queries reduce it further. Each hop's student MSE sets the reference scale (100%). Quadrature and student errors use separate banks of 100 and 96 prompts, respectively. Appendix D.1 gives the protocol.

Curve capacity and learnable curvature. Figure 8 separates offline H3 trajectory fits from linear probes on frozen student features. Raising K from 2 to 8 lifts the offline explained fraction from 46% to 90% on the strongly curved hop nearest noise and from 90% to 99% on the smoother final hop, so the curve family can represent teacher trajectories almost exactly. Student features predict far less: held-out probes explain 45–50% of final-hop variation for every $K \in \{2, 3, 4, 8\}$ and only 7–11% on the first hop. The learned curve is correspondingly a shrunken bend aligned with the teacher's (Section 4.3), which keeps the queried targets learnable. Since extra terms raise this ceiling without reducing predictability, we adopt $K{=}8$ as the default; Appendix D.3 details basis conditioning, optimization, and protocols.

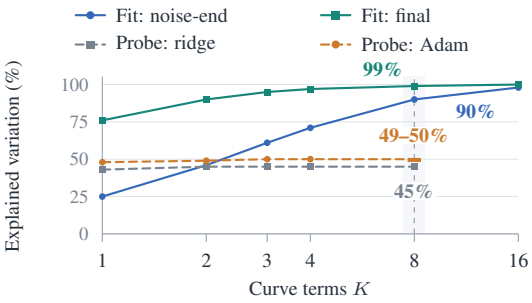

Figure 8: **Curve capacity far exceeds what student features can predict.** Offline Legendre fits (8 teacher trajectories) and final-hop linear probes (500 held-out prompts) use separate H3 banks. Solid and dashed lines denote fits and predictions, respectively. The band marks the default $K = 8$, where the Adam range spans parameterizations; probes cover $K \leq 8$. See Appendix D.3.

5 Conclusion

We presented Parametric Trajectory Distillation, approaching few-step video distillation as a capacity allocation problem. Teacher queries along student-predicted polynomial paths jointly supervise a mean-velocity output and a curvature head. This training scaffold is removed at inference, retaining the original backbone architecture and four-step sampling. Using trajectory supervision alone, PTD distills the 33B MiniMax-H3 model into a four-step LoRA student that significantly improves Astra diversity and naturalness over the state-of-the-art LightX2V Turbo while matching its dynamic and static quality. On Wan2.1-14B, PTD sets a new state of the art for trajectory distillation, outperforming PDD, the best-performing trajectory-only method on this model, under the same training setting, with significant gains in Astra dynamic quality and naturalness. Blinded human votes give PTD 63.4% and 55.1% preference shares against LightX2V Turbo and PDD. Our analyses show that the best supervision target varies along the trajectory: teacher marching on every hop softens fine detail, whereas queries along the student's own curve retain it. On the smoother final hop, we empirically found that a teacher-marched target improves detail. Although the curve family can fit teacher trajectories almost exactly, the student learns a shrunken, teacher-aligned curve that keeps its targets learnable. These results point to a simple principle for few-step video distillation: match supervision to what the student can predict.

AI USE STATEMENT

In this work, we used generative AI tools (large language model coding assistants) to write and refactor experiment, probing, and evaluation code, to draft and edit the manuscript text, and to organize the related-work survey. We used a large language model (Claude) to generate the 50 action prompts in our 100-prompt evaluation set; these prompts were fixed before video generation and were not edited after judging (Appendix C). We also used GPT-6 Astra as an automated judge for the four-dimensional video evaluation; these judgments are not human ratings. Experimental measurements were obtained by the authors, including the automated evaluation, or are explicitly attributed to prior work. All AI-assisted code was reviewed and tested by the authors, and all AI-drafted text was checked against our experimental logs by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Section 3 and Appendix B specify the hop grid, the parametric curve, the query positions and velocity least-squares objective, the backbone-specific recipes (per-hop LoRA placement and ranks for MiniMax-H3 and full fine-tuning for Wan2.1-14B), the training setting shared by PTD and our PDD reproduction, and the optimizer settings. Appendix B.4 describes removal of the curvature head and, for MiniMax-H3, adapter merging for inference. Appendix C describes evaluation prompt construction, the four-dimensional automated evaluation, the Wan2.1-14B VBench protocol, and the blinded human study; Appendix D specifies the trajectory and teacher-query diagnostics, and Appendix G gives the full judging rubric. The trained MiniMax-H3 LoRA adapters and Wan2.1-14B checkpoint, the training and probing code, the evaluation prompts, and the judging scripts will be released.

REFERENCES

Anonymous. Parallel decoding distillation for fast image and video generation. In *Submitted to The Fifteenth International Conference on Learning Representations (ICLR 2027)*, 2026. Under review.

Mahmoud Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, et al. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Nicholas M. Boffi, Michael S. Albergo, and Eric Vanden-Eijnden. Flow map matching with stochastic interpolants: A mathematical framework for consistency models. *Transactions on Machine Learning Research*, 2025a.

Nicholas M. Boffi, Michael S. Albergo, and Eric Vanden-Eijnden. How to build a consistency model: Learning flow maps via self-distillation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025b. URL https://arxiv.org/abs/2505.18825.

Hansheng Chen, Kai Zhang, Hao Tan, Leonidas Guibas, Gordon Wetzstein, and Sai Bi. π-Flow: Policy-based few-step generation via imitation distillation. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2510.14974.

Tim Dockhorn, Arash Vahdat, and Karsten Kreis. GENIE: Higher-order denoising diffusion solvers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL https://arxiv.org/abs/2210.05475.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning (ICML)*, pp. 12606–12633, 2024.

Xiangyu Fan, Zesong Qiu, Zhuguanyu Wu, Fanzhou Wang, Zhiqian Lin, Tianxiang Ren, Dahua Lin, Ruihao Gong, and Lei Yang. Phased DMD: Few-step distribution matching distillation via score matching within subintervals. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 41667–41676, 2026. URL https://arxiv.org/abs/2510.27684.

FastVideo Team. FastVideo FastH3 V1: Open-weight 4-step sparse distilled Minimax H3 for 14x speedup on NVIDIA Blackwell GPU, 2026. URL https://haoailab.com/blogs/fasth3-preview/. Technical release, August 27, 2026.

Ling Feng and Sikun Yang. Bezier distillation. *arXiv preprint arXiv:2503.16562*, 2025. URL https://arxiv.org/abs/2503.16562.

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *International Conference on Learning Representations (ICLR)*, 2025.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Zhengyang Geng, Yiyang Lu, Zongze Wu, Eli Shechtman, J. Zico Kolter, and Kaiming He. Improved mean flows: On the challenges of fastforward generative models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 30467–30476, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Geng_Improved_Mean_Flows_On_the_Challenges_of_Fastforward_Generative_Models_CVPR_2026_paper.html.

Jiatao Gu, Chen Wang, Shuangfei Zhai, Yizhe Zhang, Lingjie Liu, and Joshua M. Susskind. Data-free distillation of diffusion models with bootstrapping. In *International Conference on Machine Learning (ICML)*, pp. 16622–16646, 2024. URL https://proceedings.mlr.press/v235/gu24d.html.

Yuchao Gu, Guian Fang, Yuxin Jiang, Weijia Mao, Song Han, Han Cai, and Mike Zheng Shou. AnyFlow: Any-step video diffusion model with on-policy flow map distillation. *arXiv preprint arXiv:2605.13724*, 2026. URL https://arxiv.org/abs/2605.13724.

Yi Guo, Wei Wang, Zhihang Yuan, Rong Cao, Kuan Chen, Zhengyang Chen, Yuanyuan Huo, Yang Zhang, Yuping Wang, Shouda Liu, and Yuxuan Wang. SplitMeanFlow: Interval splitting consistency in few-step generative modeling. *arXiv preprint arXiv:2507.16884*, 2025. URL https://arxiv.org/abs/2507.16884.

Hui Han, Siyuan Li, Jiaqi Chen, Yiwen Yuan, Yuling Wu, Yufan Deng, Chak Tou Leong, Hanwen Du, Junchen Fu, Youhua Li, Jie Zhang, Chi Zhang, Li-jia Li, and Yongxin Ni. Video-Bench: Human-aligned video generation benchmark. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18858–18868, 2025.

Xuan He, Dongfu Jiang, Ge Zhang, Max Ku, Achint Soni, Sherman Siu, Haonan Chen, Abhranil Chandra, Ziyan Jiang, Aaran Arulraj, Kai Wang, Quy Duc Do, Yuansheng Ni, Bohan Lyu, Yaswanth Narsupalli, Rongqi Fan, Zhiheng Lyu, Bill Yuchen Lin, and Wenhu Chen. VideoScore: Building automatic metrics to simulate fine-grained human feedback for video generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2105–2123, 2024.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS Workshop on Deep Generative Models and Downstream Applications*, 2021.

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

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self Forcing: Bridging the train-test gap in autoregressive video diffusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Karthik Inbasekar, Guy Rom, and Omer Shlomovits. WorldJen: An end-to-end multi-dimensional benchmark for generative video models. *arXiv preprint arXiv:2605.03475*, 2026.

Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ODE trajectory of diffusion. In *International Conference on Learning Representations (ICLR)*, 2024.

Kyungmin Lee, Sihyun Yu, and Jinwoo Shin. Decoupled MeanFlow: Turning flow models into flow maps for accelerated sampling. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/699124f548e4cb86674b20eae0fd8619-Abstract-Conference.html.

Shanchuan Lin, Xin Xia, Yuxi Ren, Ceyuan Yang, Xuefeng Xiao, and Lu Jiang. Diffusion adversarial post-training for one-step video generation. In *International Conference on Machine Learning (ICML)*, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*, 2023.

Yuanxin Liu, Rui Zhu, Shuhuai Ren, Jiacong Wang, Haoyuan Guo, Xu Sun, and Lu Jiang. UVE: Are MLLMs unified evaluators for AI-generated videos? In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2025.

Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models. In *International Conference on Learning Representations (ICLR)*, 2025.

Yunhong Min, Juil Koo, Seungwoo Yoo, and Minhyuk Sung. BézierFlow: Learning Bézier stochastic interpolant schedulers for few-step generation. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2512.13255.

MiniMax. MiniMax-H3, 2026. URL https://huggingface.co/MiniMaxAI/MiniMax-H3.

ModelTC. MiniMax-H3 Turbo, 2026. URL https://github.com/ModelTC/Minimax-H3-Turbo. LightX2V four-step LoRAs; weights at https://huggingface.co/lightx2v/Minimax-h3-Turbo.

multimodalart. H3 Acceleration Arena, 2026. URL https://huggingface.co/spaces/multimodalart/h3-acceleration-arena. Leaderboard accessed September 23, 2026.

Kepan Nan, Rui Xie, Penghao Zhou, Tiehan Fan, Zhenheng Yang, Zhijie Chen, Xiang Li, Jian Yang, and Ying Tai. OpenVid-1M: A large-scale high-quality dataset for text-to-video generation. In *International Conference on Learning Representations (ICLR)*, 2025.

Weili Nie, Julius Berner, Nanye Ma, Chao Liu, Saining Xie, and Arash Vahdat. Transition matching distillation for fast video generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4645–4655, 2026. URL https://arxiv.org/abs/2601.09881.

Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, et al. Movie Gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024.

Amirmojtaba Sabour, Sanja Fidler, and Karsten Kreis. Align your flow: Scaling continuous-time flow map distillation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2506.14603.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations (ICLR)*, 2022.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning (ICML)*, 2023.

Team Wan. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Joshua Tian Jin Tee, Kang Zhang, Hee Suk Yoon, Dhananjaya Nagaraja Gowda, Chanwoo Kim, and Chang D. Yoo. Physics informed distillation for diffusion models. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=rOvaUsF996.

Shangyuan Tong, Nanye Ma, Saining Xie, and Tommi Jaakkola. Flow map distillation without data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/papers/Tong_Flow_Map_Distillation_Without_Data_CVPR_2026_paper.pdf.

Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. VideoMAE V2: Scaling video masked autoencoders with dual masking. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Wenhao Wang and Yi Yang. VidProM: A million-scale real prompt-gallery dataset for text-to-video diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2024.

Timing Yang, Sucheng Ren, Alan Yuille, and Feng Wang. ViMix-14M: A curated multi-source video-text dataset with long-form, high-quality captions and crawl-free access. *arXiv preprint arXiv:2511.18382*, 2025.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Frédo Durand, and William T. Freeman. Improved distribution matching distillation for fast image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024a.

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

Huijie Zhang, Aliaksandr Siarohin, Willi Menapace, Michael Vasilkovsky, Sergey Tulyakov, Qing Qu, and Ivan Skorokhodov. AlphaFlow: Understanding and improving MeanFlow models. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/e8c20cafe841cba3e31a17488dc9c3f1-Abstract-Conference.html.

Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. UniPC: A unified predictor-corrector framework for fast sampling of diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Dian Zheng, Ziqi Huang, Hongbo Liu, Kai Zou, Yinan He, Fan Zhang, Lulu Gu, Yuanhan Zhang, Jingwen He, Wei-Shi Zheng, Yu Qiao, and Ziwei Liu. VBench-2.0: Advancing video generation benchmark suite for intrinsic faithfulness. *arXiv preprint arXiv:2503.21755*, 2025.

Hongkai Zheng, Weili Nie, Arash Vahdat, Kamyar Azizzadenesheli, and Anima Anandkumar. Fast sampling of diffusion models via operator learning. In *International Conference on Machine Learning (ICML)*, pp. 42390–42402, 2023. URL https://proceedings.mlr.press/v202/zheng23d.html.

Kaiwen Zheng, Yuji Wang, Qianli Ma, Huayu Chen, Jintao Zhang, Yogesh Balaji, Jianfei Chen, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. Large scale diffusion distillation via score-regularized continuous-time consistency. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2510.08431.

Linqi Zhou, Mathias Parger, Ayaan Haque, and Jiaming Song. Terminal velocity matching. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2511.19797.

Mingyuan Zhou, Huangjie Zheng, Zhendong Wang, Mingzhang Yin, and Hai Huang. Score identity distillation: Exponentially fast distillation of pretrained diffusion models for one-step generation. In *International Conference on Machine Learning (ICML)*, pp. 62307–62331, 2024. URL https://proceedings.mlr.press/v235/zhou24x.html.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4590–4601, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Zou_DisCa_Accelerating_Video_Diffusion_Transformers_with_Distillation-Compatible_Learnable_Feature_Caching_CVPR_2026_paper.html.

A Additional method details

A.1 Endpoint-constrained curves and supervision

Throughout this appendix we fix a hop k and write $\boldsymbol{h}$, $\boldsymbol{c}_j$, $\boldsymbol{x}_\theta$, and $\boldsymbol{v}_\theta$ for $\boldsymbol{h}_\theta$, $\boldsymbol{c}_{\theta,j}$, $\boldsymbol{x}_{\theta,k}$, and $\boldsymbol{v}_{\theta,k}$. With $\sigma = 2\tau - 1$ the Legendre polynomials are $P_0 = 1$, $P_1 = \sigma$, $P_2 = \frac{1}{2}(3\sigma^2 - 1)$, $P_3 = \frac{1}{2}(5\sigma^3 - 3\sigma)$, and so on; we use K=8. The curvature basis functions are $b_j(\tau) = \tau(\tau - 1)P_j(2\tau - 1)$. The parameterized trajectory in equation 1 is $\boldsymbol{x}_\theta(\tau) = \boldsymbol{x}_k + \Delta_k[\tau\boldsymbol{h} + \sum_{j<K} b_j(\tau)\boldsymbol{c}_j]$. Differentiating with respect to $s = t_k + \tau\Delta_k$ divides by Δ_k and gives equation 2 with

$$d_j(\tau) = \frac{\mathrm{d}b_j(\tau)}{\mathrm{d}\tau} = (2\tau - 1)P_j(\sigma) + 2\tau(\tau - 1)\frac{\mathrm{d}P_j(\sigma)}{\mathrm{d}\sigma}. \tag{7}$$

Since $b_j(0) = b_j(1) = 0$, the curvature terms leave both endpoints fixed at a given $\boldsymbol{h}$ and satisfy $\int_0^1 d_j(\tau)\,\mathrm{d}\tau = 0$. Consequently,

$$\boldsymbol{x}_\theta(1) = \boldsymbol{x}_k + \Delta_k\boldsymbol{h}, \qquad \int_0^1 \boldsymbol{v}_\theta(\tau)\,\mathrm{d}\tau = \boldsymbol{h}. \tag{8}$$

These exact parameterization identities justify retaining $\boldsymbol{h}$ as the distilled output and removing the curvature head at inference.

Both heads receive the same residual. For fixed, detached teacher queries, define $\boldsymbol{r}_i = \boldsymbol{h} + \sum_{j<K} d_j(\tau_i)\boldsymbol{c}_j - \boldsymbol{u}_i$. Before the implementation's gradient routing (Appendix A.2), the ordinary output gradients of equation 5 are

$$\frac{\partial\mathcal{L}_{\mathrm{PTD}}}{\partial\boldsymbol{h}} = 2\sum_i \omega_i\boldsymbol{r}_i, \qquad \frac{\partial\mathcal{L}_{\mathrm{PTD}}}{\partial\boldsymbol{c}_j} = 2\sum_i \omega_i d_j(\tau_i)\boldsymbol{r}_i. \tag{9}$$

Thus a local teacher target directly supervises the endpoint output and the curve coefficients through one velocity residual. The K coefficients are latent-sized vectors emitted by one curvature head.

How the endpoint output learns a mean velocity. Fix the query curve $\boldsymbol{x}_{\theta_0}$ at the current parameters θ_0 and define the detached teacher field along it as $\widetilde{\boldsymbol{u}}(\tau) = \mathrm{sg}[u_{\mathrm{T}}(\boldsymbol{x}_{\theta_0}(\tau), s(\tau) \mid y)]$. For uniform queries, or an unbiased stratified estimator of the same integral, equation 9 gives

$$\begin{aligned}\mathbb{E}\left[\frac{\partial\mathcal{L}_{\mathrm{PTD}}}{\partial\boldsymbol{h}}\right] &= 2\int_0^1 \left[\boldsymbol{h} + \sum_{j<K} d_j(\tau)\boldsymbol{c}_j - \widetilde{\boldsymbol{u}}(\tau)\right]\mathrm{d}\tau \\ &= 2\left[\boldsymbol{h} - \int_0^1 \widetilde{\boldsymbol{u}}(\tau)\,\mathrm{d}\tau\right].\end{aligned} \tag{10}$$

Equation 10 is the output partial derivative of the unnormalized least-squares (LS) objective with query locations held fixed and teacher targets detached, averaged over query positions. Its target is the teacher velocity averaged along the current student curve; Appendix A.3 relates the local velocity residual to the teacher endpoint. H3's modality normalization subsequently rescales the sampled gradients.

Coverage across query positions. For one latent coordinate the least-squares design row is $\mathbf{z}(\tau) = [1, d_0(\tau), \ldots, d_{K-1}(\tau)]$. For m=2, the design matrix has rank at most two, providing at most two independent constraints on the nine output values for each latent coordinate. Across random query positions, the basis is fully observable: d_j is a polynomial of degree $j + 1$ with a nonzero leading coefficient, so $1, d_0, \ldots, d_{K-1}$ are linearly independent. Under a query density that is positive throughout $(0, 1)$, the expected Gram matrix $\mathbb{E}[\mathbf{z}(\tau)^\top\mathbf{z}(\tau)]$ is therefore positive definite. For our equal-weight stratified pair, the expectation of equation 5 is exactly the uniform integral of the squared residual, since the two half-intervals together cover $[0, 1]$. This provides supervision of the full profile across training draws. With $K = 8$, the position and velocity polynomials have maximum degrees nine and eight, respectively.

A.2 FINITE-QUERY GRADIENT ROUTING

Both evaluated PTD recipes use the finite-query LS value in equation 5, with gradients routed through an exact decomposition on student-curve-supervised hops. For equal weights, write $\bar{d}_j = m^{-1}\sum_i d_j(\tau_i)$, $\bar{\boldsymbol{u}} = m^{-1}\sum_i \boldsymbol{u}_i$, and

$$\bar{\boldsymbol{r}} = \boldsymbol{h} + \sum_j \bar{d}_j \boldsymbol{c}_j - \bar{\boldsymbol{u}}, \qquad \widetilde{\boldsymbol{r}}_i = \sum_j [d_j(\tau_i) - \bar{d}_j]\boldsymbol{c}_j - (\boldsymbol{u}_i - \bar{\boldsymbol{u}}). \tag{11}$$

Since $\sum_i \widetilde{\boldsymbol{r}}_i = 0$, the following equality holds for every query set:

$$\mathcal{L}_{\mathrm{PTD}} = \|\bar{\boldsymbol{r}}\|^2 + \frac{1}{m}\sum_i \|\widetilde{\boldsymbol{r}}_i\|^2. \tag{12}$$

In the first term, the implementation replaces $\sum_j \bar{d}_j \boldsymbol{c}_j$ by its stop-gradient value; we denote the resulting routed objective by $\mathcal{L}_{\mathrm{route}}$. Before per-modality loss normalization, this preserves the loss value and the output partial derivative with respect to $\boldsymbol{h}$, while changing the curvature gradient:

$$\frac{\partial \mathcal{L}_{\mathrm{route}}}{\partial \boldsymbol{h}} = 2\bar{\boldsymbol{r}}, \qquad \frac{\partial \mathcal{L}_{\mathrm{route}}}{\partial \boldsymbol{c}_j} = \frac{2}{m}\sum_i [d_j(\tau_i) - \bar{d}_j]\widetilde{\boldsymbol{r}}_i. \tag{13}$$

Relative to ordinary LS differentiation, the omitted term is $2\bar{d}_j\bar{\boldsymbol{r}}$. Because $\bar{d}_j$ and $\bar{\boldsymbol{r}}$ share the sampled query positions, the expected omitted term is $2\,\mathrm{Cov}(\bar{d}_j, \bar{\boldsymbol{r}})$, since $\mathbb{E}[\bar{d}_j] = \int_0^1 d_j(\tau)\,\mathrm{d}\tau = 0$ under the stratified queries. The shared backbone still receives gradients through both $\boldsymbol{h}$ and $\boldsymbol{c}$.

A.3 FROM LOCAL VELOCITY FITTING TO A DISTILLED ENDPOINT

The training target in PTD is the teacher's local velocity on a parameterized curve. The relation between this local supervision and the teacher's endpoint follows from the ODE residual, as in the local-dynamics analysis of flow-map distillation (Boffi et al., 2025a).

Fix a hop start $\boldsymbol{x}_k$, its conditioning, and the student's outputs. Let $\varphi_k(\boldsymbol{x}_k, s)$ denote the teacher ODE solution from $\boldsymbol{x}_k$ at t_k, and write $\boldsymbol{x}_{\mathrm{T}}(\tau) = \varphi_k(\boldsymbol{x}_k, s(\tau))$. Define

$$\boldsymbol{r}(\tau) = \boldsymbol{v}_\theta(\tau) - u_{\mathrm{T}}(\boldsymbol{x}_\theta(\tau), s(\tau)). \tag{14}$$

Assume the teacher field is Lipschitz in its state with constant L_{T} on a region containing the two paths. Both paths start at $\boldsymbol{x}_k$, and their difference satisfies

$$\frac{\mathrm{d}}{\mathrm{d}\tau}(\boldsymbol{x}_\theta - \boldsymbol{x}_{\mathrm{T}}) = \Delta_k\big[u_{\mathrm{T}}(\boldsymbol{x}_\theta, s) - u_{\mathrm{T}}(\boldsymbol{x}_{\mathrm{T}}, s) + \boldsymbol{r}(\tau)\big]. \tag{15}$$

Gronwall's inequality and Cauchy–Schwarz give

$$\begin{aligned}\|\boldsymbol{x}_k + \Delta_k\boldsymbol{h} - \boldsymbol{x}_{\mathrm{T}}(1)\| &\le |\Delta_k| \int_0^1 e^{|\Delta_k| L_{\mathrm{T}}(1-\tau)}\|\boldsymbol{r}(\tau)\|\,\mathrm{d}\tau \\ &\le |\Delta_k| e^{|\Delta_k| L_{\mathrm{T}}} \left(\int_0^1 \|\boldsymbol{r}(\tau)\|^2\,\mathrm{d}\tau\right)^{1/2}.\end{aligned} \tag{16}$$

Thus, under the Lipschitz condition, the student-curve objective directly controls the distilled endpoint: with equal-weight stratified queries and fixed outputs, the expectation of equation 5 equals $\int_0^1 \|\boldsymbol{r}(\tau)\|^2\,\mathrm{d}\tau$, which bounds the error of $\boldsymbol{x}_k + \Delta_k\boldsymbol{h}$ through equation 16 without integrating the teacher. If the residual vanishes throughout the hop, ODE uniqueness gives $\boldsymbol{x}_\theta(\tau) = \boldsymbol{x}_{\mathrm{T}}(\tau)$, and $\boldsymbol{h}$ equals the teacher trajectory's average velocity. The curvature coefficients represent this path during training; the endpoint uses only $\boldsymbol{h}$.

A.4 RELATION TO LAGRANGIAN FLOW MAPS, TVM, PDD, AND π-FLOW

This section places PTD among flow-map, terminal-velocity, interval-supervision, and policy-based distillation (Table 2). We keep our convention throughout: generation runs from noise at $t = 1$ to data at $t = 0$, a map runs from a start time t to a target time $s \le t$, and $\Phi_{t\to s}$ denotes the teacher flow map, so $\Phi_{t_k \to s(\tau)}(\boldsymbol{x}_k) = \varphi_k(\boldsymbol{x}_k, s(\tau)) = \boldsymbol{x}_{\mathrm{T}}(\tau)$ in Appendix A.3. We rewrite equations from works that order the two times differently in this convention.

Table 2: **PTD among Lagrangian and related distillation methods.** LMD: Lagrangian map distillation (Boffi et al., 2025a;b); TVM: Terminal Velocity Matching (Zhou et al., 2026); π-Flow (Chen et al., 2026); PDD (Anonymous, 2026). RK: Runge–Kutta; JVP: Jacobian–vector product. PTD's entries describe its student-curve loss; H3's final interval uses a teacher march instead (Appendix B.3).

	Velocity target	Target query states	Student velocity	Deployed update
LMD	Frozen teacher	Predicted map states	JVP of a two-time network	Two-time map
TVM	Own EMA velocity, plus flow matching on data	Predicted terminal states	JVP of a two-time network, backpropagated	Two-time map
π-Flow	Frozen teacher	Detached policy rollout, numerically integrated	Closed-form policy	Policy integrated with substeps
PDD	Teacher midpoint RK target	Teacher solver stages	Discrete interval outputs	Fused interval outputs
PTD	Frozen teacher	Detached student curve, in closed form	Analytic basis derivative	$\boldsymbol{x}_k + \Delta_k \boldsymbol{h}_\theta$

Lagrangian flow-map distillation. The teacher flow map is the unique solution of the Lagrangian equation (Boffi et al., 2025a)

$$\partial_s \Phi_{t\to s}(\boldsymbol{x}) = u_{\mathrm{T}}\big(\Phi_{t\to s}(\boldsymbol{x}), s\big), \qquad \Phi_{t\to t}(\boldsymbol{x}) = \boldsymbol{x}. \tag{17}$$

Lagrangian map distillation (LMD) fits a map $\hat{X}_{t\to s}$ with $\hat{X}_{t\to t}(\boldsymbol{x}) = \boldsymbol{x}$ by penalizing the residual of equation 17 at the map's own predicted states. In the stop-gradient form of Boffi et al. (2025b), which avoids the teacher's spatial Jacobian, the objective is

$$\mathcal{L}_{\mathrm{LMD}}(\hat{X}) = \mathbb{E}_{(t,s)\sim w}\, \mathbb{E}_{\boldsymbol{x}\sim\rho_t} \Big\| \partial_s \hat{X}_{t\to s}(\boldsymbol{x}) - u_{\mathrm{T}}\big(\mathrm{sg}[\hat{X}_{t\to s}(\boldsymbol{x})], s\big) \Big\|^2, \tag{18}$$

where w is a distribution over time pairs and ρ_t is a start-state distribution. Boffi et al. (2025a;b) draw start states from the stochastic-interpolant marginal, spread start and target times over all of $[0,1]$, and parameterize $\hat{X}_{t\to s}(\boldsymbol{x}) = \boldsymbol{x} + (s-t)\, \boldsymbol{g}_\theta(\boldsymbol{x}, t, s)$ with a network $\boldsymbol{g}_\theta$ conditioned on both times. The time derivative $\partial_s \hat{X}$ then differentiates the network and is computed with the map by a JVP, at about 1.5 times the cost of a forward pass (Boffi et al., 2025b).

PTD as a restricted Lagrangian map. For interval k and $s \in [t_{k+1}, t_k]$, let $\tau = (s - t_k)/\Delta_k$. Since $(s - t_k)(\tau - 1) = \Delta_k \tau(\tau - 1)$, the curve in equation 1 is the flow-map ansatz

$$\hat{X}^\theta_{t_k\to s}(\boldsymbol{x}_k) = \boldsymbol{x}_k + (s - t_k)\Big[\boldsymbol{h}_\theta + \sum_{j<K} (\tau - 1) P_j(2\tau - 1)\, \boldsymbol{c}_{\theta,j}\Big] = \boldsymbol{x}_{\theta,k}(\tau), \tag{19}$$

with $\boldsymbol{h}_\theta$ and $\boldsymbol{c}_\theta$ evaluated once at $(\boldsymbol{x}_k, t_k)$. The bracket is the average velocity from t_k to s, which Flow Map Matching and MeanFlow obtain from a network conditioned on both times (Boffi et al., 2025a; Geng et al., 2025); in PTD it is a polynomial in τ whose coefficients come from one evaluation at the teacher's single-time interface. Let w place the start time on a grid point t_k and draw s uniformly from $[t_{k+1}, t_k]$, and let ρ_{t_k} be the law of the detached rollout state $\boldsymbol{x}_k$. The ansatz in equation 19 then has three exact properties.

(a) $\partial_s \hat{X}^\theta_{t_k\to s}(\boldsymbol{x}_k) = \boldsymbol{v}_{\theta,k}(\tau)$, a fixed linear combination of the outputs.

(b) At (t_k, s_i), the integrand of equation 18 equals the i-th squared residual in equation 5, so the equal-weight stratified loss in equation 5 is an unbiased estimate of equation 18 on interval k, in value and in semigradient.

(c) $\hat{X}^\theta_{t_k\to t_{k+1}}(\boldsymbol{x}_k) = \boldsymbol{x}_k + \Delta_k \boldsymbol{h}_\theta$ for every $\boldsymbol{c}_\theta$, so the sampler in equation 6 composes these one-interval maps without evaluating $\boldsymbol{c}_\theta$.

For (a), $s = t_k + \tau\Delta_k$ gives $\partial_s = \Delta_k^{-1}\partial_\tau$, and differentiating equation 1 yields equation 2. For (b), substituting $s_i = s(\tau_i)$ and (a) into equation 18 gives $\|\boldsymbol{v}_{\theta,k}(\tau_i) - u_{\mathrm{T}}(\mathrm{sg}[\boldsymbol{x}_{\theta,k}(\tau_i)], s_i)\|^2 =$

$\|\boldsymbol{h}_\theta + \sum_j d_j(\tau_i)\boldsymbol{c}_{\theta,j} - \boldsymbol{u}_i\|^2$ by equation 4. One uniform draw from each half of $[0,1]$ with equal weights has expectation $\int_0^1$ of this integrand, and uniform τ is uniform s on the interval. The law of the draws does not depend on θ, so the same expectation holds for the semigradient, taken with the query states and ρ_{t_k} fixed. For (c), the factor $\tau - 1$ vanishes at $s = t_{k+1}$.

Relative to equation 18, PTD thus (i) fixes start times to the grid and keeps target times within the current interval, (ii) draws start states from its own detached rollout, (iii) places the target-time dependence in a fixed polynomial basis, and (iv) estimates the target-time integral with the stratified pair m=2. It keeps the stop-gradient placement of equation 18, which the original Lagrangian loss of Boffi et al. (2025a) does not use. Boffi et al. (2025a) state that their minimizer result holds for any start-state process with a strictly positive density, and use the interpolant so that the map is learned where it is evaluated; PTD's rollout serves the same purpose, training each map on the states its sampler visits. Physics Informed Distillation corresponds to a single interval starting at $t = 1$, with a time-conditioned network and finite-difference derivatives, and therefore generates in one step (Tee et al., 2024).

What the restrictions buy. Restrictions (i) and (iii) let the network keep the teacher's input $(\boldsymbol{x}_k, t_k)$: the target time enters only through the basis, which spans a single interval. Restriction (iii) makes (a) analytic: all m target times share one student pass and ordinary backpropagation, with no JVP. Property (c) separates the endpoint output from the interior shape: since $b_j(1) = 0$, the endpoint depends on $\boldsymbol{h}_\theta$ alone, so the curvature head can be removed without changing any deployed map, whereas a two-time map keeps its target-time input at deployment. The curve family in equation 1 contains the exact teacher segment $s \mapsto \Phi_{t_k \to s}(\boldsymbol{x}_k)$ if and only if that segment is a polynomial of degree at most K+1 in s, so PTD controls the endpoint through the residual rather than through exact representability. If the Lipschitz condition of Appendix A.3 holds for every start state in the support of ρ_{t_k}, coupling the two one-interval pushforwards of ρ_{t_k} through the shared start state and squaring equation 16 gives, with $\hat{\Phi}_k(\boldsymbol{x}_k) = \boldsymbol{x}_k + \Delta_k \boldsymbol{h}_\theta(\boldsymbol{x}_k, t_k)$,

$$W_2^2\big(\hat{\Phi}_{k\#}\rho_{t_k},\, \Phi_{t_k \to t_{k+1}\#}\rho_{t_k}\big) \le \Delta_k^2 e^{2|\Delta_k| L_{\mathrm{T}}}\, \mathbb{E}_{\boldsymbol{x}_k \sim \rho_{t_k}} \int_0^1 \|\boldsymbol{r}(\tau)\|^2 \,\mathrm{d}\tau, \tag{20}$$

where the right-hand expectation is the expected value of equation 5 at the current outputs. This is the interval-level analogue of the Lagrangian Wasserstein bound of Boffi et al. (2025a), whose exponent uses a one-sided Lipschitz constant.

Scope of the identity. The correspondence is exact for the value of equation 5 and for its ordinary semigradient in equation 9. The implemented routing (Appendix A.2) keeps the loss value and the $\boldsymbol{h}$-gradient, before H3's modality normalization, and omits $2\bar{d}_j\bar{\boldsymbol{r}}$ from the $\boldsymbol{c}$-gradient. The $\boldsymbol{h}$-component of the output gradient therefore equals that of the Lagrangian semigradient, while the curvature component follows the centered residuals. H3's final interval replaces the residual by the teacher-marched target in equation 21.

Terminal Velocity Matching. TVM uses the same time convention and writes the map as a displacement $\boldsymbol{f}(\boldsymbol{x}_t, t, s) = \Phi^{\mathrm{data}}_{t \to s}(\boldsymbol{x}_t) - \boldsymbol{x}_t$, where Φ^{data} is the flow map of the marginal velocity $\boldsymbol{u}^{\mathrm{data}}$ between data and prior (Zhou et al., 2026). Its terminal velocity condition, $\frac{\mathrm{d}}{\mathrm{d}s}\boldsymbol{f}(\boldsymbol{x}_t, t, s) = \boldsymbol{u}^{\mathrm{data}}(\Phi^{\mathrm{data}}_{t \to s}(\boldsymbol{x}_t), s)$, is equation 17 for that field, written for the displacement. TVM trains from scratch: it replaces the unknown velocity at the predicted terminal state by the model's own velocity, with stop-gradient on the state and EMA weights for the target, and adds a flow-matching loss on data, so its first term has the form of Lagrangian self-distillation (Boffi et al., 2025b). Per query, the residual in equation 5 is TVM's terminal-velocity residual with the self-generated target replaced by the frozen teacher and (t, s) restricted to $(t_k, s(\tau))$. With $\boldsymbol{f}_\theta = (s - t)\boldsymbol{F}_\theta(\boldsymbol{x}_t, t, s)$, TVM computes $\frac{\mathrm{d}}{\mathrm{d}s}\boldsymbol{f}_\theta = \boldsymbol{F}_\theta + (s - t)\partial_s \boldsymbol{F}_\theta$ with a JVP and backpropagates through it, using a fused attention kernel that supports the JVP backward pass. In PTD, the same terminal derivative is $\boldsymbol{v}_{\theta,k}(\tau)$, read from the outputs through known basis derivatives. TVM bounds W_2^2 between its one-step model distribution and the data by a weighted integral of its objective plus a constant, assuming the model velocity is Lipschitz, an assumption that motivates its architectural changes to the diffusion transformer. Equation 20 is the per-interval analogue for a frozen teacher: its residual is evaluated on the student's own curve, where equation 5 queries, its Lipschitz constant is the teacher's rather than the trained network's, and its reference distribution is the teacher's one-interval pushforward of ρ_{t_k}.

PDD. PDD predicts 128 interval-velocity outputs, or slots, on a fixed time grid, trains selected outputs against teacher midpoint Runge–Kutta block targets with block sizes 16–64, and at inference fuses four blocks of 32 slots into one endpoint projection per block (Anonymous, 2026). For both methods, the ideal target for a supervised interval is the teacher's exact mean velocity over that interval from its start state; for PTD on interval k this is $\boldsymbol{h}_k^\star = (\Phi_{t_k \to t_{k+1}}(\boldsymbol{x}_k) - \boldsymbol{x}_k)/\Delta_k$. The two methods approach this target differently. A midpoint target queries the teacher at stage states placed by the teacher's own solver steps; it is fixed for a given start state and differs from the exact mean velocity by the solver's truncation error, to which PDD's trajectory-recovery result is subject (Appendix E). PTD's targets are exact teacher velocities at states on the student's current curve, so they move with the student; the residual in equation 14 measures the remaining mismatch and bounds the endpoint error through equation 16. When the student curve equals the teacher segment, $\boldsymbol{x}_{\theta,k}(\tau) = \Phi_{t_k \to s(\tau)}(\boldsymbol{x}_k)$, every query lies on the teacher path, the residual vanishes, and equation 3 gives $\boldsymbol{h}_\theta = \boldsymbol{h}_k^\star$, the quantity that a midpoint target approximates up to its truncation error. H3's final interval is the one place where PTD uses a solver target, the eight-step Euler march in equation 21.

π-Flow. In π-Flow, one student evaluation at the interval start outputs a network-free policy $\pi(\boldsymbol{x}, s)$, and the imitation loss matches π to the teacher at states on a detached, numerically integrated rollout of the policy's own ODE (Chen et al., 2026). PTD's tangent defines such a policy, $\pi_k(\boldsymbol{x}, s) = \boldsymbol{v}_{\theta,k}\big((s - t_k)/\Delta_k\big)$, which depends on time only. Its rollout from $\boldsymbol{x}_k$ integrates in closed form to $\boldsymbol{x}_{\theta,k}(\tau)$, so the imitation residual at a rollout state is $\boldsymbol{v}_{\theta,k}(\tau_i) - \boldsymbol{u}_i$, the residual in equation 5. For this policy class, on-policy imitation and equation 18 therefore coincide. For state-dependent policies, such as π-Flow's DX and GMFlow policies, the two losses have the same value but different gradients: imitation detaches the policy's state input, while the semigradient of equation 18 differentiates through it. Both methods thus match teacher velocities along the student's own trajectory; they differ in what is deployed. π-Flow expands the output layer to emit its policy and integrates the policy with substeps at inference; PTD deploys only the policy's interval average $\boldsymbol{h}_\theta$, so each step is the update in equation 6 at the backbone's original output width.

B Implementation details

B.1 Time grids

We place student hop boundaries on the shifted grid $t = \frac{\gamma u}{1+(\gamma-1)u}$ with shift γ=6 and base time $u \in [0, 1]$ (Esser et al., 2024). The four hops are the images of the quarters $u \in \{1, \frac{3}{4}, \frac{1}{2}, \frac{1}{4}, 0\}$, giving $t = (0.999, 0.947, 0.857, 0.667, 0)$ (the first value is clamped) and $\Delta = (-0.052, -0.090, -0.190, -0.667)$. For H3, the student uses the release coupling $t^{\mathrm{aud}} = 0.25t/(1 - 0.75t)$ for audio. The LightX2V Turbo baseline retains its released video shift-6 and audio shift-3 sampler settings. On Wan2.1-14B, both students use four blocks of 32 slots from the same 128-slot shift-6 grid.

B.2 TRAINING CONFIGURATIONS

Table 3: **Configurations of the evaluated PTD checkpoints.** The two backbones use different adaptation and final-hop training recipes; CFG denotes classifier-free guidance (Ho & Salimans, 2021) and SLG skip-layer guidance.

	Wan2.1-14B	MiniMax-H3 (33B)
Height × width; frames	480×832; 81	768×1344; 124
Frame rate	16 fps	24 fps
Backbone training	Full fine-tuning	Per-hop LoRA, rank 128
Velocity projection	Dense, trainable	Per-hop LoRA
Curvature terms K	8	8
Student-curve queries	2 on every hop	2 on hops 0–2
Final-hop supervision	Routed velocity LS	8-step teacher march
Teacher guidance	CFG 5; SLG, layer 12	CFG-distilled release
Learning rate (constant)	10^{-5}	10^{-4}
Global batch	256	512
Evaluated iteration	64	150
EMA; warmup	None	None

Prompt corpora and checkpoints. H3 uses 55,879 prompts drawn from ViMix, OpenVid, and VidProM and rewritten with the official H3 rewriter into its three-field audio-video template. Text conditions are pre-encoded. Our leakage filter removes the ViMix validation captions, the source of our evaluation prompts, from this corpus. Training consumes $150 \times 512 = 76{,}800$ hop-level samples; since one rollout spans four iterations, these come from about $19{,}200$ prompt rollouts, approximately 0.34 corpus passes. Both Wan students train on randomly sampled prompts from the same ViMix prompt pool. All training uses prompts without paired target videos.

MiniMax-H3 adaptation. The teacher is the frozen CFG-distilled H3 release; one teacher query requires one conditional forward pass. The student shares this base and adds a rank-128 LoRA set per hop (Hu et al., 2022), including hop-specific LoRA on the original velocity projection. No time-embedding or adaptive-layer-normalization LoRA is used. The curvature head has eight Legendre terms. Hops 0–2 use two stratified queries, with one uniform sample in each half-interval, and the gradient routing in Appendix A.2. The last hop uses only the teacher-marched mean-velocity target in equation 21. Each modality's loss is divided by its detached value, with audio weight 1. The learning rate is 10^{-4} for adapters and heads, with gradient clipping at 1, no EMA or warmup, and fp32 carried states. The global batch is accumulated over 64 GPUs with one sample per GPU and eight accumulation rounds. The reported checkpoint is iteration 150; its output has 124 frames and stereo 32-kHz audio.

Wan2.1-14B adaptation and baseline. We reproduce PDD with the training hyperparameters reported by Anonymous (2026) and train PTD under the same training setting. Both students train on Wan2.1-14B with global batch 256 and constant learning rate 10^{-5}, without warmup, weight decay, or EMA, using bf16 computation and fp32 master weights; PTD accumulates this batch over 128 GPUs with two rounds and PDD over 64 GPUs with four. Both methods query the same frozen teacher with CFG 5, skip-layer guidance (SLG) on layer 12, and the official negative prompt. PTD trains the full transformer and dense velocity projection, together with the eight-term curvature head; the run has no LoRA parameters. Every hop uses two stratified student-curve queries with the split gradient routing of Appendix A.2, without teacher marching. PDD uses 128 slot heads, midpoint Runge–Kutta block targets, and training block sizes 16–64; inference fuses four blocks of 32 slots. PDD is trained for 256 iterations, and its checkpoints at iterations 64, 96, and 256 reach overall VBench scores of 83.29, 83.67, and 82.83 (Table 6). PTD reaches 84.57 at iteration 64, outperforming PDD both at the same iteration count and at PDD's best checkpoint, iteration 96, which serves as the baseline in the Astra, human-preference, and VBench comparisons. The Wan PTD run continues to iteration 240; iteration 64 is its best evaluated checkpoint by overall VBench score. Our PDD reproduction scores below the VBench values reported by Anonymous (2026),

which we attribute to differences in training prompt data and evaluation prompts (the published rows use Self Forcing prompts) and to VBench evaluation noise.

Rollout and update accounting. Workers process the same hop at each training iteration, carry detached student endpoints to the next hop, and restart from noise after the fourth hop. A complete rollout therefore spans four reported iterations; each iteration is one optimizer update, reported as an update in the main text.

B.3 H3 FINAL-HOP SUPERVISION

Starting from the detached carried state $\boldsymbol{x}_3$ at $t = 0.667$, the H3 recipe marches the teacher to the endpoint and trains only the mean-velocity prediction:

$$\boldsymbol{h}_{\mathrm{m}} = \frac{\boldsymbol{x}_{\mathrm{end}} - \boldsymbol{x}_3}{\Delta_3}, \qquad \mathcal{L}_{\mathrm{H3},3} = \left\| \boldsymbol{h}_\theta - \mathrm{sg}[\boldsymbol{h}_{\mathrm{m}}] \right\|^2 . \tag{21}$$

There is no student-curve velocity loss or curvature-head supervision on this hop. The eight Euler intervals use the shift-12 nodes $0.667, 0.632, 0.591, 0.541, 0.481, 0.405, 0.308, 0.180, 0$. The release's modality coupling determines the corresponding audio updates; the displayed target is per modality, with its own time interval and loss units. The endpoint depends on the carried state and teacher integration, rather than the current hop's predicted curve. This integration is used only during training. Hops 0–2 retain the two-query student-curve loss. In H3, we march only the final hop. On high-noise hops the teacher trajectory bends sharply, and a marched endpoint folds this curvature into a target that is too hard for the student to learn in one evaluation; marching every hop softens local detail on Wan2.1-1.3B (Section 4.4). The final hop is smoother: with $K{=}2$ Legendre terms, offline fits explain 90% of its velocity variation, versus 46% on the hop nearest noise (Figure 8). We empirically found that marching this hop alone improves H3 visual detail at approximately 20% higher training compute than the all-two-query recipe (Section 4.2).

Model-specific recipes. The H3 checkpoint adds final-hop marching to the student-curve objective, whereas the Wan checkpoint uses the student-curve objective alone on all four hops.

Modality units. H3 follows the release's coupled audio/video time parameterization and modality-specific update rules. Each modality loss is normalized by its detached value before their equally weighted combination. Wan has a single modality and does not use this normalization.

B.4 REMOVING THE TRAINING SCAFFOLD

The curvature head is unnecessary for the endpoint in equation 8 and is removed at inference. On H3, a hop's adapted projection, including the mean-velocity output, can be materialized as

$$W^{(k)} = W + B^{(k)} A^{(k)}, \tag{22}$$

with adapter scale 1. The low-rank product has the original projection's shape, so merging preserves the original layer types and output dimensions. The shared frozen-base representation with active hop adapters computes the same linear maps in exact arithmetic, with different storage requirements. On Wan, the fully fine-tuned backbone and dense mean-velocity projection are retained directly; there are no adapters to merge. Both samplers take four steps using $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \Delta_k \boldsymbol{h}_\theta(\boldsymbol{x}_k, t_k)$, without evaluating curvature coefficients. The deployed models therefore retain their original architecture and latent output interface.

C EVALUATION PROTOCOL AND ADDITIONAL RESULTS

Models and scoring dimensions. The primary comparisons are PTD at 150 updates against the released LightX2V Turbo v1.1 four-step LoRA on MiniMax-H3, and PTD at 64 updates against our PDD reproduction at 96 updates on Wan2.1-14B. All students use four network evaluations and seeds 0, 1, and 2 for each prompt. Turbo retains its released sampler settings and LoRA scale 1. GPT-6 Astra judges the outputs with medium reasoning effort and the V5 rubric reproduced in Appendix G. The four dimensions are dynamic quality (motion realism), static quality (individual-frame quality), naturalness, and diversity across the three seeds. Greater motion alone does not improve the dynamic-quality score. Audio is not evaluated.

Evaluation prompt construction. The 100-prompt evaluation set comprises two equally weighted groups, selected without judging model outputs. V50 contains 50 held-out ViMix-14M (Yang et al., 2025) captions of real videos. We shuffle the 804 eligible validation captions with seed 20260922, discard captions shorter than 15 words, and expand candidates using the public H3 prompt rewriter. In that order, we retain the first 50 rewrites that pass the official-template check and begin with a live-action tag; accepted rewrites are used verbatim. Our leakage filter excludes these source captions from the prompts used to train our H3 student. D50 contains 50 Claude-generated, medium-dynamics, single-shot action prompts, with five prompts per category across ten categories. These prompts were fixed before generation; they describe one subject performing a continuous physical action, with natural lighting and secondary motion, without cuts or stylization. No prompts were added, removed, or edited after judging. For H3, the judge receives the full official template prompt. For Wan, we retain the video-description field and remove template headers, audio fields, and shot tags; the resulting prompts have median length 142 words. Each model pair yields 600 clips and 400 judge calls over the 100 prompts.

Frames, blinding, and repetition. Each call receives 12 timestamped images, sampled uniformly over the clip including its first and last frame, in temporal order. Each image is a 2×3 mosaic: one model per row and seeds 0, 1, and 2 across the columns. The row labels show only position and seed, concealing model identities from the judge. For H3, timestamps are 0.00, 0.46, 0.92, 1.42, 1.88, 2.33, 2.79, 3.25, 3.71, 4.21, 4.67, and 5.12 seconds; each video cell is resized to 256×448 pixels (height $\times$ width). The Wan evaluation uses the same sampling protocol on its 16-fps clips. Each row ordering is judged twice, giving four calls per prompt, and rows are mapped back to model identities before aggregation. Output follows a JSON schema with keys `evidence`, `top_row`, `bottom_row`, and `preference`; evidence identifies relevant frames and seeds.

Score aggregation. Each call assigns one integer score from 1 to 4 per model and dimension, judging the three clips jointly. We average the four scores within each prompt, then average across prompts with equal weights. We compute paired score differences within prompts and bootstrap prompts to obtain 95% confidence intervals; the H3 summaries use 4,000 percentile-bootstrap resamples with random seed 0. The bootstrap unit is the prompt, with its seeds, time points, and repeated calls grouped together. Reported differences use unrounded per-prompt scores and can differ slightly from subtracting the rounded model means. We report each scoring dimension separately, aggregating scores across both row orders. The automated ratings use the sampled video frames; human preference is evaluated through full-video playback.

Main-figure aggregation. For the four automated rows in Figure 4, let d_p be the mean of the four paired score differences for prompt p. A PTD win has $d_p \geq 0.25$, a loss has $d_p \leq -0.25$, and a tie has $|d_p| < 0.25$. Writing these prompt counts as W, T, and L, the figure assigns half a credit to each method for every tie: PTD occupies $(W + T/2)/N$ of the bar and the baseline occupies $(L + T/2)/N$, where $N = 100$. These shares are derived from prompt-level scores, not the judge's direct preference labels. Table 4 preserves the mean scores, paired confidence intervals, and original win/tie/loss counts.

Results by prompt group. Table 5 separates the two prompt groups of the 100-prompt evaluation. Every significant gain in Table 4 holds within each prompt group: the paired 95% intervals exclude zero in both V50 and D50 for H3 diversity ($+1.24$ and $+1.28$) and naturalness ($+0.19$ and $+0.04$), and for Wan dynamic quality ($+0.29$ and $+0.24$) and naturalness ($+0.19$ and $+0.25$).

Wan2.1-14B VBench. We reproduce PDD (Anonymous, 2026) strictly following its reported training hyperparameters, with the same backbone, frozen teacher and guidance, ViMix training prompts, global batch size 256, and constant learning rate 10^{-5} as PTD (Appendix B). We train PDD for 256 updates and evaluate checkpoints at 64, 96, and 256 updates; update 96 has the highest overall score and serves as the PDD baseline for VBench, Astra, and the human study (Table 6). PTD reaches its best evaluated checkpoint after only 64 updates, where it already exceeds all three PDD checkpoints in both overall and quality scores; at the matched 64 updates, it leads PDD by 1.28 overall points (84.57 vs. 83.29). Against the best evaluated PDD checkpoint, PTD scores higher on 13 of the 16 VBench dimensions, with the largest gains in spatial relationship ($+8.73$), multiple objects ($+8.72$), and scene ($+5.67$) (Table 7). VBench dynamic degree measures motion amount, whereas

Table 4: **GPT-6 Astra scores on 100 evaluation prompts.** Mean scores on a 1–4 scale; δ is the paired mean difference (PTD minus baseline), with 95% bootstrap confidence intervals over 100 prompts. Each prompt has three seeds per model and four judge calls. W/T/L gives score-derived wins, ties, and losses from PTD's perspective.

Dimension	PTD	Baseline	δ	95% CI	W/T/L
MiniMax-H3: PTD at 150 updates vs. released LightX2V Turbo v1.1					
Dynamic quality	2.70	2.61	$+0.09$	$[-0.04, +0.21]$	45/27/28
Static quality	3.04	3.06	-0.03	$[-0.08, +0.02]$	6/86/8
Naturalness	3.09	2.98	$+0.11$	$[+0.06, +0.17]$	28/68/4
Diversity	2.81	1.55	$+1.26$	$[+1.15, +1.36]$	93/6/1
Wan2.1-14B: PTD at 64 updates vs. our PDD reproduction at 96 updates					
Dynamic quality	2.43	2.16	$+0.27$	$[+0.17, +0.36]$	47/41/12
Static quality	2.85	2.84	$+0.01$	$[-0.06, +0.07]$	12/78/10
Naturalness	2.82	2.60	$+0.22$	$[+0.13, +0.32]$	44/44/12
Diversity	2.47	2.52	-0.04	$[-0.17, +0.09]$	36/22/42

Model means and paired differences are rounded independently; their displayed differences may disagree by 0.01. W/T/L is derived from the paired prompt-level score difference using a 0.25 threshold.

Table 5: **Astra results by prompt group.** Mean paired differences (PTD minus baseline) with 95% bootstrap confidence intervals over prompts. H3 uses LightX2V Turbo v1.1 as the baseline; Wan compares PTD at 64 updates with our PDD reproduction at 96 updates.

Prompt group	n	Dynamic quality	Static quality	Naturalness	Diversity
MiniMax-H3					
Live-action (V50)	50	$+0.07$ $[-0.09, +0.23]$	-0.06 $[-0.16, +0.04]$	$+0.19$ $[+0.09, +0.29]$	$+1.24$ $[+1.09, +1.39]$
Action (D50)	50	$+0.10$ $[-0.10, +0.30]$	$+0.00$ $[+0.00, +0.00]$	$+0.04$ $[+0.01, +0.07]$	$+1.28$ $[+1.12, +1.43]$
Wan2.1-14B					
Live-action (V50)	50	$+0.29$ $[+0.14, +0.45]$	$+0.00$ $[-0.10, +0.10]$	$+0.19$ $[+0.06, +0.33]$	-0.12 $[-0.30, +0.07]$
Action (D50)	50	$+0.24$ $[+0.12, +0.38]$	$+0.01$ $[-0.06, +0.10]$	$+0.25$ $[+0.12, +0.38]$	$+0.03$ $[-0.15, +0.21]$

Positive differences favor PTD. Dynamic quality rewards motion realism, not motion amount. All values, including zero-width intervals, retain the source precision.

Astra's dynamic quality measures motion realism; PTD significantly outperforms PDD in dynamic quality (Table 4). As in Section 4.1, we report VBench as a reference; Astra ratings and blinded human votes are our primary evidence. Our VBench runs use 946 prompts with five seeds and four network evaluations, whereas the published rows of Table 1 use Self Forcing prompts (Huang et al., 2025), except for rCM's proprietary prompts.

Embedding-based diversity. We report V-JEPA 2 (Assran et al., 2025) and VideoMAE V2 (Wang et al., 2023) cosine-distance diversity scores on the five-seed VBench samples, following the metric convention of Anonymous (2026). Larger distances indicate greater variation in the encoder's feature space. These scores are separate from the VBench aggregates and from Astra's prompt-conditioned diversity rating on the 100 evaluation prompts.

Human preference study. Ten raters participated in blinded pairwise comparisons of PTD with LightX2V Turbo v1.1 on MiniMax-H3 and with our PDD reproduction on Wan2.1-14B. The final analysis contains 161 votes: 82 on H3 and 79 on Wan. The checkpoints match the automated evaluation: PTD at 150 updates on H3, and PTD at 64 versus PDD at 96 updates on Wan; all

Table 6: **Wan2.1-14B checkpoints on VBench.** PDD is trained for 256 updates and evaluated at 64, 96, and 256 updates; update 96 has the highest overall score and is the best evaluated PDD checkpoint. PTD is shown at its best evaluated checkpoint, 64 updates, where it exceeds all three PDD checkpoints in overall and quality scores. Bold marks the best overall score. All use four network evaluations.

Method	Updates	Overall	Quality	Semantic
PDD	64	83.29	84.43	78.71
PDD (best evaluated)	96	83.67	85.24	77.40
PDD	256	82.83	83.21	81.29
PTD (best evaluated)	64	**84.57**	85.44	81.10

Table 7: **Detailed VBench results for the best evaluated checkpoints.** PDD uses update 96 and PTD update 64; both use four network evaluations. Higher is better under the VBench convention. Bold marks the higher score; δ is PTD minus PDD in score points.

Dimension	PDD @96	PTD @64	δ
Quality dimensions			
Subject consistency	94.45	**95.79**	+1.34
Background consistency	95.51	**96.34**	+0.83
Temporal flickering	98.74	**98.80**	+0.06
Motion smoothness	**98.86**	98.37	−0.49
Dynamic degree	**85.83**	76.11	−9.72
Aesthetic quality	63.45	**66.73**	+3.28
Imaging quality	66.75	**68.40**	+1.65
Semantic dimensions			
Object class	94.68	**96.11**	+1.43
Multiple objects	80.21	**88.93**	+8.72
Human action	**97.60**	97.40	−0.20
Color	82.87	**86.81**	+3.94
Spatial relationship	70.56	**79.29**	+8.73
Scene	49.10	**54.77**	+5.67
Appearance style	20.56	**20.95**	+0.39
Temporal style	24.28	**24.61**	+0.33
Overall consistency	26.35	**26.88**	+0.53
Quality	85.24	**85.44**	+0.20
Semantic	77.40	**81.10**	+3.70
Overall	83.67	**84.57**	+0.90

Differences are calculated from the displayed scores. Dynamic degree measures motion amount. Each checkpoint is its method's best evaluated checkpoint by overall VBench score.

generators use four network evaluations. The study reuses the videos from the automated evaluation. Each comparison contains 99 eligible prompts from that evaluation: 49 live-action captions and 50 action prompts, after one live-action prompt was excluded before data collection. For each prompt, the interface presents two rows of three clips using matching seeds 0, 1, and 2, with model names and file identifiers concealed. Raters choose the row they would rather watch, or a tie. Picture quality, motion quality, naturalness, and diversity guide this single holistic preference.

Each session interleaves 20 items, comprising ten per model comparison and five from each prompt subset within each comparison. The server prioritizes items with fewer votes and avoids assigning prompts already judged by the same rater. Top/bottom placement is balanced per item with random tie-breaking, and presentation order is shuffled. Clips play together and loop muted; an enlarged H3 clip can play audio. The interface requests one full playback before voting, with a timeout fallback for browser playback events. Raters may revisit an item; only the latest vote per session and item is retained.

Each retained vote contributes equally. Let W, T, and L denote PTD wins, ties, and losses. We summarize each comparison by the preference share $(W+T/2)/(W+T+L)$, splitting ties equally as in the automated bars of Figure 4. On H3, PTD receives 45 wins, 14 ties, and 23 losses against LightX2V Turbo, a 63.4% preference share. On Wan, it receives 33 wins, 21 ties, and 25 losses against PDD, a 55.1% preference share. PTD thus wins more votes than it loses in both comparisons. Table 8 gives the action and live-action breakdowns together with non-tie win rates, Wilson 95% confidence intervals, and two-sided exact sign-test p-values.

Table 8: **Human preference results.** n counts votes. Share is PTD's preference share with ties split equally, as in Figure 4; the non-tie win rate excludes ties. We report Wilson 95% confidence intervals for the non-tie win rate and two-sided exact sign-test p-values.

Subset	n	PTD wins	Ties	Losses	Share	Non-tie win rate	95% CI	p
MiniMax-H3: PTD at 150 updates vs. LightX2V Turbo v1.1								
All	82	45	14	23	63.4%	66%	[54, 76]	0.010
Action (D50)	43	22	10	11	62.8%	67%	[50, 80]	0.08
Live-action (V50)	39	23	4	12	64.1%	66%	[49, 79]	0.09
Wan2.1-14B: PTD at 64 updates vs. PDD at 96 updates								
All	79	33	21	25	55.1%	57%	[44, 69]	0.36
Action (D50)	40	21	8	11	62.5%	66%	[48, 80]	0.11
Live-action (V50)	39	12	13	14	47.4%	46%	[29, 65]	0.85

D Additional diagnostics

D.1 Accuracy of two-query target estimation

Protocol. The reported training recipe uses two stratified queries, one drawn independently and uniformly from each half of a hop. We assess this sampling rule and a three-query alternative on a bank of 100 H3 teacher trajectories, stored at 64 integration steps on the shift-12 clock. The diagnostic uses the training hop boundaries $[0.999, 0.947, 0.857, 0.667, 0]$ and interpolates the stored teacher velocities within each hop. For each query count $m \in \{2, 3\}$, we draw 32 independent stratified query sets per trajectory and form

$$\bar{\boldsymbol{u}}_m = \frac{1}{m}\sum_{r=1}^{m} \boldsymbol{u}(\tau_r), \qquad \tau_r \overset{\text{ind.}}{\sim} \mathcal{U}\left[\frac{r-1}{m}, \frac{r}{m}\right). \tag{23}$$

We compare this estimate with the reference mean velocity obtained from the stored endpoint displacement divided by the signed hop duration. The squared discrepancy is normalized by the reference mean-velocity energy. This is a teacher-path diagnostic of the query rule; training itself queries the student's predicted path. We report video hops 0–2, the H3 hops trained with student-curve queries; the final hop is trained with the teacher-marched target instead (Appendix B.3).

Two queries give a small error relative to the prediction residual. Table 9 compares the normalized quadrature discrepancy with the mean-velocity error of the reported H3 checkpoint after 150 updates. This student error is measured on 96 separate teacher-bank prompts, with the student started from the teacher-path state at each hop start, as the squared relative L_2 error of its predicted mean velocity with respect to the teacher's reference mean velocity on that hop. The two-query discrepancy is 12.6%, 13.6%, and 13.3% of this student error across the three hops, or approximately one eighth (Figure 7). Three queries reduce the ratios to 6.6%, 6.1%, and 5.8%, respectively, at one additional teacher evaluation per supervised hop. The training recipe therefore retains two queries.

D.2 Within-hop trajectory learning on Wan2.1-14B

Protocol. We examine the learned trajectory shapes of PTD and our PDD reproduction on eight VBench prompts, using a shift-6 time grid and a 128-slot teacher reference path. The comparison

Table 9: **Two- and three-query estimation errors relative to student prediction error.** Video mean-velocity errors on H3 hops 0–2, normalized by the reference velocity energy.

Hop	Student MSE	Two queries		Three queries	
		MSE	Relative	MSE	Relative
0	0.3346	0.04218	12.6%	0.02224	6.6%
1	0.0114	0.00155	13.6%	0.00069	6.1%
2	0.0036	0.00048	13.3%	0.00021	5.8%

All MSEs are normalized squared errors; relative values divide by the corresponding student MSE. Quadrature: 100 teacher trajectories, 32 random query sets per trajectory and query count. Student: reported H3 checkpoint (150 updates), squared teacher-anchored relative L_2 mean-velocity error on a separate 96-prompt bank. Relative values compare the quadrature and student error scales measured on their respective banks.

covers checkpoints at 32–224 student updates; PTD at 240 and PDD at 256 updates are listed separately as unmatched endpoints. Checkpoints are matched by student-update count. This diagnostic concerns latent trajectory shape and is separate from both VBench video scores and the 100-prompt Astra evaluation.

Bend magnitude and direction. Let $\mathbf{b}_S$ and $\mathbf{b}_T$ denote the student and teacher bend vectors measured by the probe in the middle half of each hop, $\tau \in [0.25, 0.75]$. For nonzero vectors, their magnitude ratio, directional alignment, and teacher-aligned projection are

$$r = \frac{\|\mathbf{b}_S\|}{\|\mathbf{b}_T\|}, \qquad \cos\theta = \frac{\langle \mathbf{b}_S, \mathbf{b}_T \rangle}{\|\mathbf{b}_S\|\,\|\mathbf{b}_T\|}, \qquad q = \frac{\langle \mathbf{b}_S, \mathbf{b}_T \rangle}{\|\mathbf{b}_T\|^2}. \tag{24}$$

The projection q is an unbounded, signed coefficient that isolates the component of the student's bend parallel to the teacher's bend, normalized by teacher bend energy. For a single vector pair, $q = r\cos\theta$; we average each metric independently over the probe samples. Tables 10 and 11 give each statistic separately.

Calibration ratio. We also report $r/\cos\theta$. For a vector pair with positive inner product, its reciprocal is the scalar α that minimizes $\|\alpha\mathbf{b}_S - \mathbf{b}_T\|^2$, giving the least-squares calibration of the predicted bend magnitude.

Trajectory shape across hops. The teacher bends most in the high-noise first hop: on the eight probe prompts, its velocity direction turns by 43.5° within hop 0, compared with 4.7° within the final hop. In this strongly curved hop, PTD has a larger magnitude ratio r, direction cosine $\cos\theta$, and teacher-aligned projection q than PDD at every shared checkpoint from 32 to 224 updates. At 64 updates, r is 0.081 for PTD and 0.024 for PDD, with cosines of 0.171 and 0.139 and projections of 0.014 and 0.003. The best evaluated PTD checkpoint (64 updates) also exceeds the best evaluated PDD checkpoint (96 updates) on all three first-hop metrics (r: 0.081 vs. 0.039; $\cos\theta$: 0.171 vs. 0.155; q: 0.014 vs. 0.006). On the smoother final hop, the teacher-aligned projections of the two methods differ by at most 0.005 at every shared checkpoint; the final hop is also where the H3 recipe applies teacher marching (Section 4.2). Tables 10 and 11 report the full progression of magnitude, direction, and projection for all four hops.

D.3 Curve expressiveness and frozen-feature prediction

The diagnostics separate fitting a known teacher trajectory from predicting its velocity variation using a student's fixed features. Figure 8 summarizes the coefficient-count comparison; Tables 12 and 13 give the basis and optimization details.

Offline trajectory fitting. Within each hop, we subtract the teacher's mean velocity and fit the remaining variation with K derivative fields d_j. Weighted least squares uses integration-step widths as weights and a double-precision QR solve. We report the fraction of squared velocity-deviation energy explained by the fit, averaged over trajectories. Thus, 99% explained means a 1% residual in

Table 10: **Wan2.1-14B trajectory-shape probe: Hops 0 and 1.** Summaries on eight VBench prompts. r: bend magnitude ratio; $\cos\theta$: bend-direction alignment; q: teacher-aligned bend projection. PTD and PDD share the listed update counts in the upper block; the final checkpoints in the lower block are unmatched.

		Hop 0				Hop 1			
Updates	Method	r	$\cos\theta$	q	$r/\cos\theta$	r	$\cos\theta$	q	$r/\cos\theta$
32	PTD	0.047	0.200	0.010	0.234	0.030	0.020	0.001	1.485
	PDD	0.017	0.139	0.002	0.123	0.025	0.023	0.001	1.065
64	PTD	0.081	0.171	0.014	0.476	0.046	0.028	0.001	1.668
	PDD	0.024	0.139	0.003	0.172	0.025	0.023	0.001	1.090
96	PTD	0.113	0.182	0.021	0.621	0.058	0.031	0.002	1.879
	PDD	0.039	0.155	0.006	0.253	0.026	0.016	0.000	1.578
128	PTD	0.138	0.226	0.033	0.613	0.059	0.035	0.002	1.673
	PDD	0.054	0.160	0.009	0.339	0.027	0.026	0.001	1.008
160	PTD	0.183	0.205	0.040	0.895	0.069	0.041	0.003	1.691
	PDD	0.043	0.180	0.008	0.239	0.026	0.022	0.001	1.200
192	PTD	0.185	0.241	0.046	0.768	0.071	0.035	0.003	2.028
	PDD	0.053	0.208	0.011	0.257	0.035	0.020	0.001	1.723
224	PTD	0.203	0.253	0.054	0.799	0.079	0.033	0.003	2.351
	PDD	0.065	0.208	0.014	0.315	0.038	0.017	0.001	2.271
Unmatched final checkpoints									
240	PTD	0.226	0.300	0.072	0.751	0.075	0.033	0.003	2.261
256	PDD	0.075	0.235	0.018	0.321	0.039	0.023	0.001	1.710

r measures bend amplitude, $\cos\theta$ its direction, and q its teacher-aligned projection. The sensitivity of $r/\cos\theta$ increases as $|\cos\theta|$ approaches zero. Bold: 64 updates, the best evaluated PTD checkpoint; the best evaluated PDD checkpoint is at 96 updates.

Table 11: **Wan2.1-14B trajectory-shape probe: Hops 2 and 3.** Summaries on eight VBench prompts. r: bend magnitude ratio; $\cos\theta$: bend-direction alignment; q: teacher-aligned bend projection. PTD and PDD share the listed update counts in the upper block; the final checkpoints in the lower block are unmatched.

		Hop 2				Hop 3			
Updates	Method	r	$\cos\theta$	q	$r/\cos\theta$	r	$\cos\theta$	q	$r/\cos\theta$
32	PTD	0.045	0.045	0.002	0.990	0.072	0.149	0.011	0.482
	PDD	0.022	0.017	0.000	1.313	0.070	0.182	0.013	0.383
64	PTD	0.049	0.051	0.003	0.953	0.130	0.166	0.022	0.780
	PDD	0.025	0.009	0.000	2.632	0.130	0.204	0.027	0.636
96	PTD	0.050	0.053	0.003	0.953	0.168	0.178	0.031	0.941
	PDD	0.033	0.019	0.001	1.700	0.160	0.214	0.035	0.748
128	PTD	0.059	0.051	0.003	1.154	0.197	0.184	0.038	1.067
	PDD	0.033	0.012	0.000	2.729	0.187	0.218	0.042	0.860
160	PTD	0.064	0.052	0.004	1.216	0.217	0.197	0.045	1.102
	PDD	0.039	0.011	0.000	3.431	0.208	0.222	0.047	0.934
192	PTD	0.072	0.052	0.004	1.397	0.247	0.205	0.053	1.205
	PDD	0.038	0.019	0.001	2.052	0.225	0.230	0.053	0.978
224	PTD	0.072	0.051	0.004	1.420	0.277	0.209	0.061	1.325
	PDD	0.035	0.019	0.001	1.882	0.244	0.234	0.058	1.043
Unmatched final checkpoints									
240	PTD	0.072	0.049	0.004	1.471	0.291	0.211	0.064	1.380
256	PDD	0.037	0.024	0.001	1.540	0.256	0.237	0.062	1.080

r measures bend amplitude, $\cos\theta$ its direction, and q its teacher-aligned projection. The sensitivity of $r/\cos\theta$ increases as $|\cos\theta|$ approaches zero. Bold: 64 updates, the best evaluated PTD checkpoint; the best evaluated PDD checkpoint is at 96 updates.

velocity-deviation energy. The H3 coefficient sweep uses eight 128-step video trajectories and the shift-6 hop grid; its values appear in Figure 8.

Basis comparison. A separate H3 bank contains 1,000 trajectories integrated with 64 Euler steps on the native shift-12 clock. Offline family comparisons use 20 fitting and 20 held-out trajectories; PCA modes are estimated only on the fitting set. Only ten integration steps fall in the final hop, so the offline coefficient-count fits use the 128-step bank instead; the 1,000-trajectory bank serves the basis comparison and the frozen-feature probes. Equivalent polynomial coordinates agree in offline fit to the reported precision. The cubic B-spline and data-driven variants are listed separately because their spaces and coefficient counts differ from the $K{=}8$ polynomial comparison.

Frozen-feature prediction. We freeze an H3 student checkpoint trained for 150 updates with student-curve supervision on a shift-12 quarter grid. Final-layer features are computed at the start of the final hop ($t = 0.81$), using teacher states from the 1,000-trajectory bank. A linear head predicts the teacher velocity deviation at three randomly sampled positions on each teacher path, with 500 prompts for fitting and 500 held out. We compare a closed-form ridge fit with a zero-initialized head optimized by Adam $(0.9, 0.99)$ for 40 batches of 250 prompts, with four optimizer steps per batch. The reported explained fraction is

$$E = 1 - \frac{\|\widehat{\delta \boldsymbol{u}} - \delta \boldsymbol{u}\|^2}{\|\delta \boldsymbol{u}\|^2} = 2r\cos\theta - r^2, \tag{25}$$

where r is the predicted-to-target amplitude ratio and θ is their angle at the sampled positions. Across the tested $K \in \{2, 3, 4, 8\}$ and every parameterization in Table 12, final-hop scores remain near 45% for ridge and 49–50% for Adam at 10^{-3}, so the Legendre default needs no basis-specific tuning at this learning rate. At 10^{-4}, coordinate choices produce larger finite-budget differences (Table 12). The final hop is also by far the most predictable: held-out scores on the earlier, higher-noise hops are 7–11%, 14–15%, and approximately 3.5% for hops 0, 1, and 2, respectively. Together with the offline fits, in which $K{=}2$ already explains 90% of final-hop variation versus 46% near the noise end (Figure 8), these scores show that the final hop is smoother than the noise-end hop and more predictable than all earlier hops, which motivates applying teacher marching only on the final hop in the H3 recipe (Section 4.2; Appendix B.3).

Table 12: **Basis choice: conditioning and finite-budget optimization.** H3 final-hop linear-probe scores (%). Natural-scale and unit-RMS Legendre coordinates are distinguished.

Parameterization	K	$\kappa(G)\downarrow$	Ridge ↑	Adam ↑	
				10^{-3}	10^{-4}
Polynomial coordinates, $K = 8$					
Legendre, natural scale	8	7.9	45	50	46
Legendre, unit RMS	8	6.3	45	50	48
Integrated Legendre, unit RMS	8	1.05	45	49	48
Bernstein (Bézier), unit RMS	8	792	45	50	49
Additional parameterizations; coefficient counts differ					
Clamped cubic B-spline	4	8.4	45	50	49
Data modes (PCA)	3	–	45	50	47

Adam scores are measured after the same 40-batch budget. G is the population Gram matrix of the derivative fields. The additional parameterizations use the coefficient counts listed in the table. –: not measured.

Population conditioning. Let $\widetilde{\mathbf{d}}(\tau)$ contain the K derivative fields with the scaling used by a given probe. We report $\kappa(G)$ for

$$G = \mathbb{E}\left[\frac{1}{2}\sum_{i=1}^{2} \widetilde{\mathbf{d}}(\tau_i)\widetilde{\mathbf{d}}(\tau_i)^{\mathsf{T}}\right], \tag{26}$$

with one query position sampled uniformly from each half of the hop. Natural-scale and unit-RMS coordinates are distinguished in Table 12; the coefficient sweep uses unit-RMS fields. This expected

Gram matrix equals the uniform-query population Gram matrix. Its condition number describes conditioning across the query distribution (Appendix A.1).

Table 13: **Additional coefficient-count diagnostics.** Population conditioning and H3 final-hop frozen-feature prediction.

K	Legendre, unit RMS $\kappa(G)\downarrow$	H3 probe (%) ↑ Adam 10^{-4}
1	1.0	39
2	1.0	40–42
3	1.6	44–48
4	2.2	45–49
8	**6.3**	**46–49**

Adam ranges are across the evaluated parameterizations after 40 batches. Bold marks the default K.

E Extended related work

We expand the comparisons in Section 2; Appendix A.4 gives a formal comparison with Lagrangian flow maps, TVM, PDD, and π-Flow.

Finite-interval mappings. Consistency models predict a common trajectory endpoint, while consistency trajectory models extend the mapping to pairs of times (Song et al., 2023; Kim et al., 2024). Flow Map Matching unifies consistency models, consistency trajectory models, and progressive distillation as learning the two-time flow map $\Phi_{s,t}$ of the underlying generative dynamics (Boffi et al., 2025a). Exact flow maps satisfy the semigroup property $\Phi_{u,t} \circ \Phi_{s,u} = \Phi_{s,t}$, so two consecutive jumps can be replaced by one (Boffi et al., 2025b). Progressive distillation trains one student step to match two teacher steps (Salimans & Ho, 2022), and shortcut models condition on the step size and train one step to match two of their own consecutive half-size steps, a discretized semigroup condition (Frans et al., 2025; Boffi et al., 2025b). Building on Flow Map Matching, Boffi et al. (2025b) convert distillation schemes into self-distillation and organize them into Eulerian, Lagrangian, and Progressive families; Align Your Flow scales continuous-time flow-map distillation (Sabour et al., 2025). MeanFlow parameterizes the flow map by the interval-average velocity and trains it with a differential identity whose target requires a Jacobian-vector product (Geng et al., 2025). Its follow-ups recast this objective as instantaneous-velocity regression through an average-velocity network (Geng et al., 2026), analyze and rebalance its trajectory-matching and consistency components (Zhang et al., 2026), replace the identity with an algebraic interval-splitting consistency that needs no Jacobian-vector products (Guo et al., 2025), convert pretrained flow models into flow maps (Lee et al., 2026), and use a restricted MeanFlow variant for stable distillation of large video models (Zou et al., 2026). PTD's $\boldsymbol{h}$ is the interval-average velocity learned by MeanFlow, but its targets come from the frozen teacher rather than from self-consistency or Jacobian-vector products of the student (Appendix A.1).

Trajectory parameterizations and local supervision. Physics Informed Distillation parameterizes the trajectory as a time-conditioned network and approximates its derivative with numerical differences between student predictions (Tee et al., 2024). DSNO learns continuous trajectories with Fourier temporal operators from solver-generated states and identifies a compact temporal spectrum in teacher trajectories (Zheng et al., 2023). π-Flow predicts a network-free velocity policy that imitates teacher velocities along detached policy rollouts and is integrated through substeps at inference (Chen et al., 2026). Transition Matching Distillation reuses backbone features while repeatedly evaluating a lightweight flow head, trained through distribution matching (Nie et al., 2026). PDD supervises selected interval outputs with teacher Runge–Kutta targets and fuses them into one endpoint projection per inference block; its trajectory-recovery result is subject to the Runge–Kutta approximation error (Anonymous, 2026). GENIE, BézierFlow, and Bézier Distillation use higher-order or polynomial structure in solver derivatives, sampling schedulers, and interpolation paths, respectively (Dockhorn et al., 2022; Min et al., 2026; Feng & Yang, 2025). PTD instead predicts

sample-dependent coefficients of an endpoint-preserving polynomial curve and deploys only its displacement output.

Distribution-level supervision. DMD and DMD2 match generated distributions through score-based supervision, and SiD optimizes a distributional discrepancy through score identities (Yin et al., 2024b;a; Zhou et al., 2024). rCM regularizes consistency training with a score-based long-skip term, and Phased DMD applies distribution matching within subintervals; both report diversity gains relative to their DMD comparisons (Zheng et al., 2026; Fan et al., 2026).

Training states and released H3 accelerators. BOOT and FreeFlow study data-free distillation through bootstrapped trajectories and flow-map learning, respectively, and DMD2 and π-Flow also train on student-generated states (Gu et al., 2024; Tong et al., 2026; Yin et al., 2024a; Chen et al., 2026). For MiniMax-H3, FastH3 releases four-step DMD2 LoRA adapters, and LightX2V Turbo, our H3 baseline, provides four-step H3 adapters (FastVideo Team, 2026; ModelTC, 2026).

Evaluating few-step video generators. On Wan2.1-14B, every published four-step student in Table 1 scores above the 50-step teacher in overall VBench, and these students differ by sub-point margins, consistent with reports that VBench saturates on strong models (Zheng et al., 2025; He et al., 2024; Inbasekar et al., 2026). We therefore use blinded human votes and GPT-6 Astra judgments as primary evidence and report VBench for reference.

F VISUAL PROMPT FOR THE FIRST-PAGE EXAMPLES

Figure 1 compares two seeds for the snowboarding prompt below, using the 50-step MiniMax-H3 teacher, four-step LightX2V Turbo v1.1, and four-step PTD. Each row shows five matching timestamps: 0.50, 1.71, 2.50, 3.42, and 4.75 seconds. All displayed frames use the same center crop, retaining 80% of the original width and height.

F.1 SNOWBOARDING

integrated_multimodal_description: [Shot 1] Live-action, cinematic, a low tracking shot follows a snowboarder in a dark grey shell jacket and a white helmet carving down a wide, freshly groomed slope under a clear sky at fast speed; “PTD” is printed in white block letters across the back of the jacket. The camera tracks forward at fast speed and shakes slightly, a few metres behind the board, as snow spray fans off every turn, the corduroy streaks past in a blur and the rider drops a gloved hand to brush the snow through a deep carve, the far ridgeline swinging across the frame. [Shot 2] At 00:02.800, the shot cuts to a wide static shot beside the slope as the rider lays the board over into a hard turn, a smooth arc of powder spraying up like a drawn curve, and the camera pans right with large amplitude at fast speed to follow the board straightening out down the fall line.

G GPT-6 ASTRA EVALUATION RUBRIC

The automated evaluation uses `RUBRIC_PAIR_V5` from `astra_judge.py`, selected by the default `--rubric v5`, with GPT-6 Astra at `medium` reasoning effort. Each call receives 12 chronologically ordered 2×3 mosaic frames, with one model per row and seeds 0/1/2 across the columns. For each prompt and model pair, each of the two row orderings is judged twice, giving four calls in total. The output is constrained by a JSON schema. Below, `{prompt}` is replaced by the full generation prompt: the official audio-video template for H3 and the extracted video description for Wan. The paper reports the four rubric scores and prompt-level outcomes derived from their differences; the preference labels requested below are not used.

You are a strict, objective judge comparing two text-to-video models on the same prompt. Each image is one time step (timestamp in the corner, in temporal order) of a 2x3 mosaic: the TOP row shows model A's three clips (seeds 0/1/2, left to right) and the BOTTOM row shows model B's three clips; the coloured bands only mark the row position and seed. Judge only visible evidence; the PROMPT is the target specification. Do NOT run any shell commands or tools.

For EACH row, score four metrics with an integer 1-4, judging the row as a whole (all three clips):

- dynamic_quality: how REAL the movement itself looks, judged against how this action and camera move would look in genuine footage. Do NOT reward more motion: a larger or faster movement is not better; the right amount is whatever the action calls for. Assess four things: (a) natural kinematics: bodies, animals, vehicles and objects move with weight, momentum, balance and follow-through (gait cycles, lean into turns, recoil, settling), not floating, gliding, sliding or robotic; (b) realistic amplitude and tempo: the size and speed of the movement match the action and the physics of the scene; exaggerated, hyperactive or jerky motion is a defect, and so is damped, sluggish, half-frozen motion; (c) no stiffness: subjects should not hold a fixed pose while the camera or background moves, repeat the same pose across frames, or look puppet-like; secondary motion (hair, cloth, water, spray, dust, shadows) exists and follows the primary motion; (d) continuity of movement and camera: trajectories are smooth and continuous, the camera move is executed as one coherent path; no jitter, strobing, teleporting, sudden reversals, foot-skating, limbs or objects deforming while they move, or identity and appearance drifting over time; motion blur is consistent with the speed shown. Infer movement from how poses, positions and blur change between consecutive frames: plausible intermediate poses with consistent displacement = fluid; identical poses over a moving background = stiff; large unexplained jumps = broken. Judge the movement only, not image sharpness, detail or colour. 4 = movement indistinguishable from real footage on all four points; 3 = minor tells (a brief floaty or stiff moment, slight blur mismatch); 2 = clearly unnatural in places (puppet-like, exaggerated or damped, sliding, jerky camera); 1 = movement largely lifeless or broken (slideshow, chaotic deformation, impossible physics).

- static_quality: image quality of the individual frames, judged as still pictures: clarity and fine detail, legible textures and text, correct anatomy and geometry, coherent structure, absence of in-frame artifacts (smearing, ghosting, extra or missing limbs, warped or disintegrating structure, blotches). Fine natural texture and film grain are not defects; over-sharpening and high contrast are not detail. Do not judge motion or temporal consistency here. 4 = no noticeable issues in any frame; 3 = minor; 2 = significant; 1 = severe.

- naturalness: how much the footage looks like real, professionally shot video (or, for stylised prompts, like a coherent production in the requested style): natural colour and tonal rendition (saturation, contrast, skin tones, highlight roll-off), believable materials, textures, lighting and depth, natural grain and motion blur, overall look and feel. Over-saturated, neon-tinted, over-sharpened, plastic or waxy "AI look" frames are defects, and so are washed-out, murky, flat or low-fidelity frames. 4 = indistinguishable from real footage / the requested style; 3 = minor tells; 2 = clearly synthetic; 1 = severe.

- diversity: how different the row's three clips are from each other in composition, subject appearance, camera and motion while all still fitting the prompt: 1 = near-identical, 2 = minor variations, 3 = clearly different in several aspects, 4 = distinct interpretations.

Then give a preference per axis (dynamic_quality, static_quality, naturalness, diversity) and overall: "top", "bottom" or "tie". Put timestamped evidence in "evidence" (name the specific frames/seeds that decide each axis). Return exactly one JSON object with keys evidence, top_row, bottom_row, preference.

PROMPT:
{prompt}