SIMPLIFYING AND IMPROVING ON-POLICY DISTILLATION FOR FEW-STEP DIFFUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Few-step distillation reduces the sampling cost of diffusion and flow models, but existing methods often rely on auxiliary score models or specialized architectures. In this work, we show that a simple on-policy regression baseline already produces high-quality and diverse samples in 4 NFE by matching teacher velocities along student rollouts, with the student velocity held constant within each sampling interval. To counter the optimization instability caused by the high-curatured velocity field of diffusion teacher, we introduce *Pivot proxy*, a lightweight residual network used only during training to model local velocity variations separately from the student interval mean prediction. We further use weaker predictions from the same teacher as negative references in a repulsive regularizer, which effectively avoid the bad modes from original policy and improve the quality. The resulting framework, *Simplified Matching Distillation (SMD)*, remains entirely regression-based, requires neither online score estimation nor adversarial training, and leaves the generator architecture and sampling procedure unchanged. We validate SMD by conducting extensive experiments on image generation and editing, 3D generation, video synthesis, and joint audio–video generation, with models of up to 33 billion parameters. Using only 4 NFE, SMD outperforms strong few-step baselines and even surpasses its multi-step teachers on several benchmarks, while naturally extending to cross-architecture and multi-teacher distillation. Our anonymous project page is available at this link.

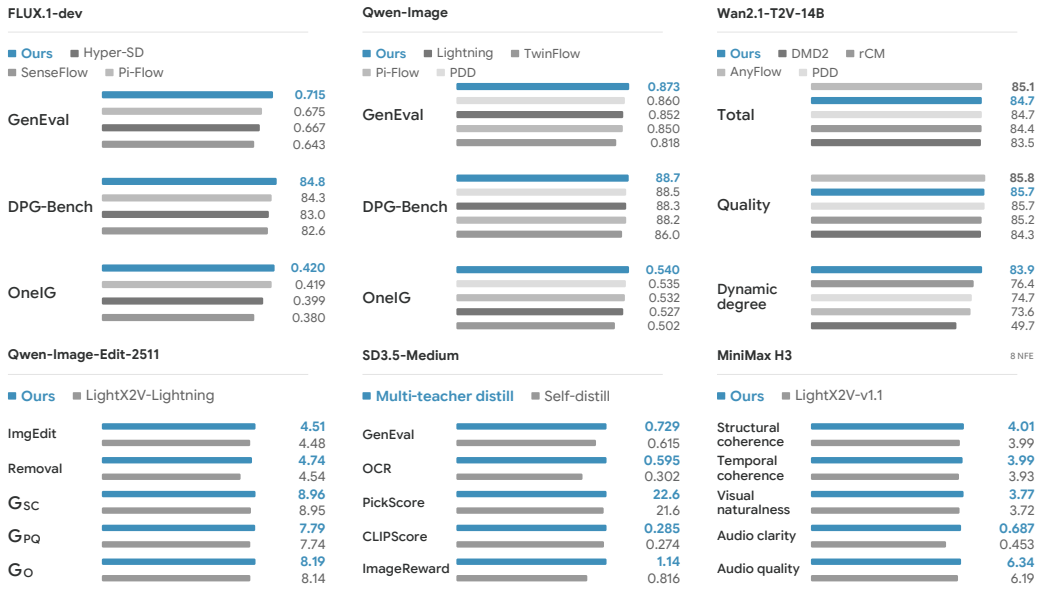


Figure 1: **Simplicity scales.** A single on-policy regression framework delivers strong few-step performance across image generation and editing, video synthesis, and joint audio–video generation, while enabling one student to learn from multiple teachers. SMD achieves this breadth without auxiliary score models, adversarial training, or changes to the student's inference architecture. Blue denotes SMD and gray denotes the comparison methods; all metrics are higher-is-better.

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

Diffusion and flow models (Ho et al., 2020; Song et al., 2021b; Lipman et al., 2023; Liu et al., 2023) have become a foundation for image, video, and 3D generation (Rombach et al., 2022; Team, 2025a;c;b; Xiang et al., 2026). Their exceptional generation quality, however, comes with a heavy inference burden, as the inherent denoising sampling typically requires tens to hundreds of sequential function evaluations. This high computational overhead thus creates a significant bottleneck for real-world deployment and interactive applications, making inference efficiency an urgent priority. Few-step distillation (Yin et al., 2024b; Salimans & Ho, 2022; Song et al., 2023; Chen et al., 2026; Shaul et al., 2026) is the way to address this challenge. By compressing a pretrained teacher model, the denoising process could be significantly reduced to a few steps, enabling rapid generation without sacrificing, and sometimes even enhancing the original generation quality (Yin et al., 2024a). Consequently, few-step distillation now has become an important and indispensable stage to post-train a frontier diffusion foundation model.

Among existing distillation paradigms, on-policy distillation, which trains a few-step student on states generated by the student itself, has emerged as the dominant approach for accelerating diffusion models. Distribution Matching Distillation (DMD) (Yin et al., 2024b) is no doubt standing as a pillar of this paradigm and has inspired a series of follow-up works (Yin et al., 2024a; Ge et al., 2026; Yin et al., 2025; Liu et al., 2026). By re-noising on-policy student generated samples, DMD approximates a reverse-KL gradient using the score difference between the teacher and the student, where the student score is dynamically tracked by an online fake model (Yin et al., 2024b). Despite its remarkable progress, DMD and its variants often suffer from low-diversity, unstable training dynamics and over-saturated output. To mitigate these issues, recent trajectory-based methods such as π-Flow (Chen et al., 2026) and PDD (Shaul et al., 2026) simplify training by directly supervising student rollouts with teacher predictions within each sampling interval. Although this removes the need for online score estimation, the few-step generator must still represent the dense velocity field within each interval, relying on either a learned policy (Chen et al., 2026) or a parallel decoder (Shaul et al., 2026), which inevitably introduces additional architectural complexity.

In this paper, we present a surprising yet compelling finding: the simplest on-policy baseline already serves as a remarkably powerful few-step diffusion distillation method. Our approach requires no auxiliary fake score model, architectural modification, or elaborate policy design. Instead, it simply matches the student velocity predictions to the teacher velocities queried at the same states, while keeping the student velocity constant within each sampling interval. Despite its extreme minimalism, this baseline consistently achieves exceptional generation quality with only 4 NFE, delivering photorealistic fidelity while preserving high-diversity across challenging benchmarks.

Finding 1. Direct velocity matching between student and teacher prediction is already a strong baseline for few-step diffusion distillation.

Building on this finding, we further apply this simple baseline to a variety of diffusion teachers (Black Forest Labs, 2024; Team, 2025a;c). However, when distilling video generation models, we observe that training dynamics can occasionally suffer from instability, as shown in Figure 3. Through the curvature analysis on the velocity field, we observe actually within a coarse integration step, the dense teacher velocities can vary substantially (Lee et al., 2023), especially for modeling complex and high-dimensional modality like video, hence exposing a single student prediction to varying targets over an entire interval to calculate the imitation loss will make the optimization conflict and jitter. To address this issue, we introduce *Pivot proxy*, a lightweight, training-only residual network that captures local velocity variations via a proxy path. By decoupling local velocity modeling from the overall student prediction, *Pivot proxy* stabilizes training while incurring no inference overhead. To further correct the student errors, we exploit weaker predictions from the same teacher as informative negative references and introduce a repulsive regularization term, which steers the student away from these inferior predictions without compromising faithful imitation of the teacher target. We term the resulting framework *Simplified Matching Distillation* (SMD). True to its name, SMD remains entirely regression-based: it requires no modification to the architecture or inference pipeline of the few-step generator, and involves neither score estimation nor adversarial objectives.

Finding 2. Pivot proxy path and repulsive term introduced by SMD further improve the stability and quality of few-step diffusion distillation.

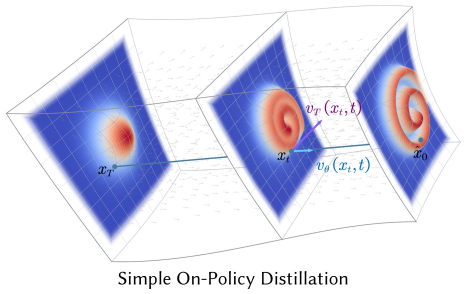


Figure 2: **Simple on-policy distillation.** The student predicts one velocity v_θ at x_t and matches it to teacher velocities v_T queried at intermediate states x_τ along its coarse transition. Algorithm 1 gives the rollout loss.

Algorithm 1: Simple on-policy distillation

Input: Student v_θ, frozen teacher v_T, condition c; grid $T = t_0 > \cdots > t_N = 0$.
Output: Rollout loss $\mathcal{L}$ for updating θ.

Sample $x \sim \mathcal{N}(0, I)$;
$\mathcal{L} \leftarrow 0$;
for $i = 0, \ldots, N-1$ **do**
  $x \leftarrow \mathrm{stopgrad}(x)$;
  $\Delta \leftarrow t_i - t_{i+1}$;
  $v_\theta \leftarrow v_\theta(x, t_i, c)$;
  Sample $\tau \sim \mathcal{U}(0, 1)$;
  $s \leftarrow t_i - \tau\Delta$;
  $x_\tau \leftarrow x - \tau\Delta v_\theta$;
  $v_T \leftarrow \mathrm{stopgrad}\big(v_T(x_\tau, s, c)\big)$;
  $\mathcal{L} \leftarrow \mathcal{L} + \|v_\theta - v_T\|_2$;
  $x \leftarrow x - \Delta v_\theta$;
return $\mathcal{L}/N$;

Such simplicity makes SMD both practical and scalable. Trained purely on-policy without any target data, SMD applies to image generation and editing, 3D generation, video synthesis, and joint audio–video generation, scaling to models with up to 33B parameters. With only 4 NFE, SMD distilled FLUX.1-dev (Black Forest Labs, 2024) and Qwen-Image (Team, 2025a) models outperform all compared few-step baselines on GenEval and DPG-Bench (Figure 1), with the FLUX.1-dev (Black Forest Labs, 2024) student even surpassing its 50-NFE teacher on both. On Wan2.1-14B (Team, 2025c), SMD achieves the highest Dynamic Degree among all compared methods, far exceeding DMD2 (Yin et al., 2024a) (83.9 vs. 49.7). Beyond acceleration, the same objective supports cross-architecture and multi-teacher distillation, where a single 4-NFE student outperforms its 28-NFE base model on all six reported metrics. These results position SMD as a powerful and versatile tool for post-training diffusion foundation models.

Finding 3. SMD shows great potentials on posttraining diffusion foundation models across different modalities and scales.

2 Method

We use v_T for the frozen teacher velocity, including its native guidance rule, and v_θ for the student. Sampling follows a decreasing grid from noise at $t = 1$ to data at $t = 0$. Background and parameter conversions are provided in Appendix B.

2.1 A Simple On-Policy Baseline

Let us set aside the complexities of existing distillation methods for a moment and start with a simple idea. We match the student's velocity to the teacher's local predictions at states along the student's own coarse transition. Despite the simplicity of both the state construction and the matching objective, this formulation already enables effective few-step distillation. Building on this observation, we introduce *Pivot*, which allows the training path to vary within each student transition while preserving its endpoints, and a repulsive objective that uses weaker predictions from the same teacher to complement velocity matching. These two simple, effective refinements require neither distribution matching nor dense trajectory alignment. Both are used only during training, preserving the student's few-step sampling regime and inference architecture. We refer to this simple on-policy formulation as *minimal distillation*.

For condition c, consider one transition from the current student rollout state x_t to time $t - \Delta$, where $\Delta > 0$. The student makes a single prediction at the transition start,

$$v_\theta = v_\theta(x_t, t, c), \qquad x_{t-\Delta} = x_t - \Delta v_\theta. \tag{1}$$

For $\tau \sim \mathcal{U}(0, 1)$, we construct an intermediate state along this transition,

$$x_\tau = x_t - \tau\Delta v_\theta, \qquad s = t - \tau\Delta. \tag{2}$$



Figure 3: **Simple distillation already works.** Fixed samples across 4-NFE training checkpoints show rapid improvement for Qwen-Image (left) but less consistent progress for Wan2.1-T2V-1.3B (right).

We then query the pretrained teacher velocity v_T at (x_τ, s, c). The same student prediction v_θ is used both to construct x_τ and to match the teacher's local velocity; no additional student evaluation is needed at the intermediate state. We optimize the L_2 distance,

$$\mathcal{L} = \mathbb{E}_\tau \left[\| v_\theta - v_T(x_\tau, s, c) \|_2 \right]. \tag{3}$$

Teacher predictions are detached and used as fixed targets. Figure 2 illustrates this construction.

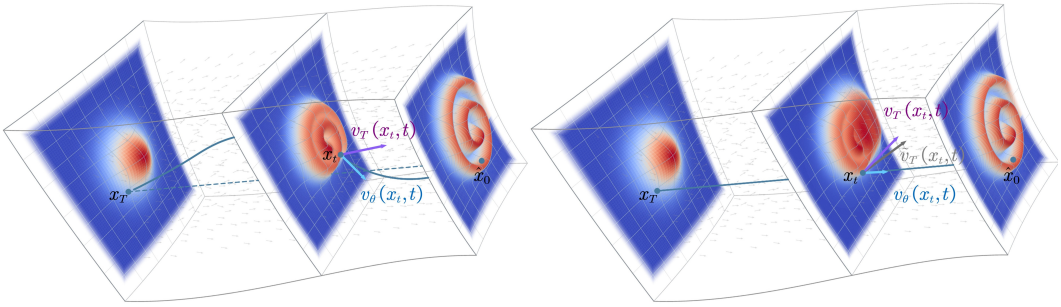
The equations above describe a single transition within a few-step student rollout. During training, we sample different conditions and initial noises, roll out the current student over the few-step grid, and apply the same objective to each transition along the rollout. Algorithm 1 summarizes the rollout loss, with gradients stopped across transitions and through teacher predictions.

Figure 3 shows fixed samples across training checkpoints for two 4-NFE students. Qwen-Image shows rapid early improvements in visual clarity and scene structure. Wan2.1-T2V-1.3B also improves substantially from initialization, but its samples exhibit pronounced quality fluctuations, with visible degradation followed by recovery across checkpoints. These observations show that local velocity matching alone can yield substantial improvements in few-step generation, although progress is not uniformly consistent across the two models.

Analysis. The results above show that local velocity matching alone can already yield substantial improvements in few-step generation. With teacher initialization, the student starts from a pretrained velocity predictor. The objective then refines this prediction using teacher evaluations at intermediate states along the student's own coarse transition. In this way, a prediction initially trained to describe local dynamics is adapted to account for the teacher velocities over a finite sampling interval.

Figure 4 illustrates successive training updates in synthetic velocity fields. At each update, we query the teacher at intermediate states on the current student transition and use its detached predictions to update the student velocity. The updated transition then determines the query state for the next teacher evaluation.

When the teacher trajectory has low curvature and its velocity varies moderately over the interval, accurate local teacher predictions provide largely compatible supervision for refining the coarse transition endpoint. This allows the student to improve its sampling behavior without an explicit endpoint target, offering a plausible explanation for the rapid improvement of Qwen-Image in Figure 3. For trajectories with higher curvature, however, the teacher velocity can change substantially in direction within a single transition. The resulting regression targets may favor conflicting updates to the student, making optimization less stable. This offers a possible explanation for the degradation and recovery observed with Wan2.1-T2V-1.3B, consistent with prior observations of large trajectory curvature in its high-noise regime (Nie et al., 2026). This analysis suggests room for improvement



(a) Imitation from a Proxy Path (b) Repulsion from a Weak Prediction

Figure 5: **Refining the on-policy baseline.** (a) A proxy path allows the local velocity to vary within a coarse transition while preserving its endpoints for a fixed v_θ. (b) Repulsion from a weaker prediction $\widetilde{v}_T$ complements imitation of the target teacher prediction v_T. Both refinements are used only during training.

in both how the student fits the teacher dynamics and how the teacher guides the student toward high-quality generation.

2.2 Imitation from a Proxy Path

To accommodate variations in the teacher velocity within a coarse transition, we introduce a proxy path that shares the endpoints of the student transition while allowing its local velocity to vary (Figure 5(a)). Specifically, we augment the coarse student velocity with a time-dependent residual,

$$v(\tau) = v_\theta + r_\phi(\tau), \tag{4}$$

subject to

$$\int_0^1 r_\phi(\tau)\, d\tau = 0. \tag{5}$$

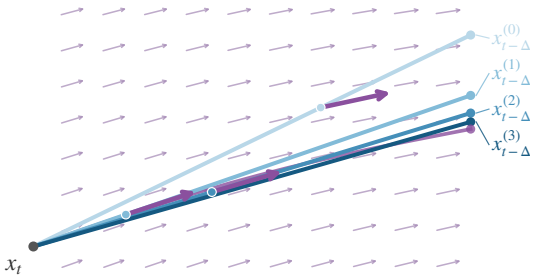
The corresponding proxy state is obtained by integrating this local velocity,

$$x_\tau = x_t - \Delta \int_0^\tau v(r)\, dr, \qquad s = t - \tau\Delta. \tag{6}$$

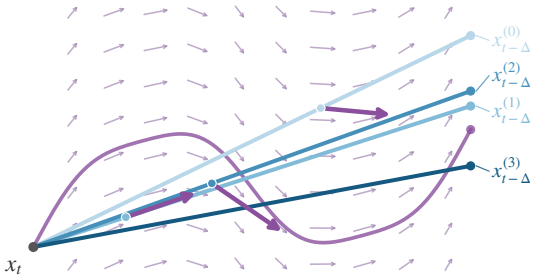
The zero-integral constraint ensures that the proxy path starts at x_t and ends at $x_t - \Delta v_\theta$. Thus, the residual can model local velocity variations without introducing an additional displacement at the end of the transition. We query the teacher at the proxy states and match its predictions with the corresponding local velocity,

$$\mathcal{L} = \mathbb{E}_\tau \left[\| v(\tau) - v_T(x_\tau, s, c) \|_2 \right]. \tag{7}$$

The student and the residual predictor are optimized jointly. The residual adjusts both the intermediate query states and the local velocities used for imitation, while the coarse student prediction continues to determine the endpoint. At inference, the student directly takes the coarse update in Eq. (1), without evaluating the residual predictor. Setting $r_\phi \equiv 0$ recovers the simple baseline.



(a) Low curvature



(b) High curvature

Figure 4: **Curvature shapes on-policy updates.** Three updates in low-curvature (top) and high-curvature (bottom) velocity fields. Blue student steps darken across updates; purple arrows and curves denote teacher queries and trajectories.

Implementation. A standard way to represent a time-dependent function is through a finite basis expansion (Ramsay & Silverman, 2005). We adopt this representation for the velocity residual and choose basis functions with zero integral to preserve the transition endpoints,

$$r_\phi(\tau) = \sum_{m=1}^{P} a_m b_m(\tau), \qquad \int_0^1 b_m(\tau)\, d\tau = 0, \tag{8}$$

where each a_m has the same shape as the student velocity. The coefficient vectors are predicted once per transition. Together with the basis functions, they determine the local velocities and proxy states throughout the interval. Pivot predicts the coefficient vectors from intermediate student features once per transition. We denote its feature-tap position, counted backward from the student end, by K, and its number of Transformer blocks by M. The branch architecture, shared computation, and initialization are described in Appendix C.

The representation does not require a particular basis. For example, a polynomial path parameterization uses

$$B_m(\tau) = \tau(1-\tau)(2\tau-1)^{m-1}, \qquad b_m(\tau) = B_m'(\tau). \tag{9}$$

Since $B_m(0) = B_m(1) = 0$, its derivative satisfies the zero-integral constraint. We explore this and other parameterizations in our ablations, with full definitions in Appendix D.1.

2.3 Repulsion from a Reference Prediction

We further refine the distillation signal using an additional prediction from the same teacher as a negative reference. We complement imitation of the target prediction with repulsion from this reference (Figure 5(b)). We retain $v_T(x_\tau, s, c)$ as the imitation target and denote the reference prediction at the same state and time by $\widetilde{v}_T(x_\tau, s, c)$. Its construction depends on the teacher's conditioning and inference procedure, as described below. We optimize

$$\mathcal{L} = \mathbb{E}_\tau \left[\|v(\tau) - v_T(x_\tau, s, c)\|_2 - \lambda \|v(\tau) - \widetilde{v}_T(x_\tau, s, c)\|_2 \right], \qquad 0 \le \lambda < 1. \tag{10}$$

The first term matches the target teacher prediction, while the second encourages separation from the reference prediction. Setting $\lambda = 0$ recovers imitation alone.

Constructing reference predictions. The repulsive objective does not require a particular construction of $\widetilde{v}_T$. For teachers using classifier-free guidance, an unconditional or conditional prediction can serve as the negative reference, corresponding to reference guidance scales $w = 0$ and $w = 1$, respectively, while v_T remains the guided prediction used for imitation. For embedded-guidance models, we use a lower guidance input instead. For models with guidance already distilled into their weights, we can instead obtain $\widetilde{v}_T$ through an additional forward pass with a layer skipped. These choices use the same pretrained model and adapt the negative reference to its inference procedure.

2.4 Training and Inference

We train student and Pivot jointly using Eq. (10) on the student's own rollout. The next rollout state is obtained from the student's coarse update; the proxy path is used only to construct supervision within the current transition. We stop gradients across transition boundaries and through teacher predictions, while retaining gradients through the student and Pivot predictions in each transition. Algorithm 2 in Appendix C summarizes the loss computation for one rollout. Losses are averaged over transitions and batch samples for joint optimization.

At inference, only the student is evaluated. Starting from Gaussian noise, it follows the same coarse updates on the same time grid, as summarized in Algorithm 3. Neither teacher prediction nor residual evaluation is required. The student architecture and number of function evaluations are unchanged.

3 Few-step Distillation

We evaluate SMD on models with up to 33B parameters across image generation, image editing, 3D generation, video synthesis, and joint audio–video generation. These tasks test whether the same distillation framework can preserve diverse generative capabilities under a limited sampling budget.

Table 1. 4-NFE image generation with FLUX.1-dev and Qwen-Image. Overall and Div. denote OneIG overall and diversity scores; † marks quoted PDD results (Shaul et al., 2026).

FLUX.1-dev					
Method	NFE	GenEval	DPG	Overall	Div.
Teacher	50	0.664	84.2	0.430	0.239
Hyper-SD	4	0.667	83.0	0.399	0.185
SenseFlow	4	0.643	82.6	0.380	0.156
Pi-Flow	4	0.675	84.3	0.419	**0.229**
Ours	4	**0.715**	**84.8**	**0.420**	0.209

Qwen-Image					
Method	NFE	GenEval	DPG	Overall	Div.
Teacher	50×2	0.874	88.3	0.540	0.195
Lightning	4	0.852	88.3	0.527	0.108
TwinFlow	4	0.818	86.0	0.502	0.152
Pi-Flow	4	0.850	88.2	0.532	0.180
PDD†	4	0.860	88.5	0.535	**0.192**
Ours	4	**0.873**	**88.7**	**0.540**	0.144

Table 3. 4-NFE Wan2.1 video generation evaluated with VBench. DD and MS denote Dynamic Degree and Motion Smoothness; † marks quoted PDD results (Euler at 1.3B).

Method	Wan2.1-T2V-1.3B				Wan2.1-T2V-14B			
	Quality	Semantic	DD	MS	Quality	Semantic	DD	MS
DMD2	84.68	79.39	45.00	98.81	84.33	80.26	49.72	**98.76**
rCM	84.61	79.56	**81.94**	97.64	85.19	81.21	76.39	97.81
AnyFlow	85.40	**81.47**	68.33	98.29	**85.78**	**82.10**	73.61	98.10
PDD†	**85.99**	78.22	72.50	**99.02**	85.69	80.71	74.72	98.52
Ours	85.64	78.97	79.44	98.66	85.72	80.52	**83.89**	98.52

Table 4. H3 structural coherence, temporal coherence, naturalness, and audio clarity/quality over 100 paired samples per budget. Bold marks larger means, not significance.

Method	8 NFE					4 NFE				
	Struct.	Temp.	Nat.	Clarity	Quality	Struct.	Temp.	Nat.	Clarity	Quality
DMD v1.1	3.985	3.925	3.720	0.453	6.190	**3.815**	3.795	3.805	0.449	6.180
Ours	**4.005**	**3.990**	**3.765**	**0.687**	**6.342**	3.805	**3.820**	**3.845**	**0.718**	**6.610**

Unless noted otherwise, students use 4 NFE. We compare against existing few-step methods using task-specific benchmarks, and include multi-step teacher results as references where available. Quantitative evaluations measure generation quality and prompt adherence, while qualitative comparisons illustrate how distillation affects visual detail, instruction following, and temporal coherence. Experimental settings and complete results are provided in Appendices H and E.

Image generation. We first test whether few-step distillation preserves prompt alignment and diverse outputs in text-to-image generation. We distill FLUX.1-dev (Black Forest Labs, 2024) and Qwen-Image (Team, 2025a) with prompt-only on-policy training. For FLUX.1-dev, we compare with Hyper-SD (Ren et al., 2024), SenseFlow (Ge et al., 2026), and Pi-Flow (Chen et al., 2026). For Qwen-Image, we compare with Lightning (ModelTC Team, 2025), TwinFlow (Cheng et al., 2026), Pi-Flow, and PDD (Shaul et al., 2026) (Table 1). Our students perform favorably on GenEval and DPG-Bench, with OneIG overall scores close to Pi-Flow on FLUX and to the teacher on Qwen. Figures 9 and 10 in the appendix show repeated samples from each method. Compared with the DMD-based SenseFlow and Lightning baselines, our students better preserve sample diversity. Direct coarse updates avoid numerical integration error from Pi-Flow's policy substeps. Compared with PDD, we train with substantially smaller batches, using a global batch size of 16 rather than 2048 for Qwen-Image.

Table 2. 4-NFE Qwen-Image-Edit-2511 comparison with the full-budget Teacher as reference.

Method	NFE	ImgEdit	SC	PQ	O
Teacher	40	4.51	8.30	8.20	7.88
Lightning	4	4.48	8.95	7.74	8.14
Ours	4	**4.51**	**8.96**	**7.79**	**8.19**

Image editing. Image editing requires following modification instructions while preserving relevant content from the source image. For Qwen-Image-Edit-2511, source images and instructions condition on-policy rollouts, without target edited images as supervision. We compare with the Lightning checkpoint (LightX2V Team, 2026). Our student improves the ImgEdit and GEdit scores (Table 2). The GEdit columns report semantic consistency (SC), perceptual quality (PQ), and overall quality

(O). In the building-to-tree example, Lightning produces more regular, stylized foliage, whereas our result blends more naturally with the surroundings (Figure 11 in the appendix).

3D shape and texture. We next test whether the same objective transfers from image latents to representations of 3D geometry and appearance. Hunyuan3D 2.1 (Hunyuan3D Team et al., 2025) uses a fixed set of shape latent vectors, while TRELLIS.2 (Xiang et al., 2026) uses spatially structured sparse latents. For Hunyuan3D, we distill the shape model. For TRELLIS.2, we distill only the final high-resolution shape and texture stages, leaving sparse-structure and coarse-shape generation unchanged. Each distilled stage uses 4 NFE. Qualitative comparisons show recognizable geometry and appearance, whereas directly reducing the teacher budget can produce incomplete shapes. Figures 12 and 13 in the appendix show results from both models. These experiments test applicability across latent representations rather than establish a quantitative 3D ranking.

Video generation. Video distillation must preserve temporal coherence and motion in addition to per-frame quality. We distill Wan2.1 (Team, 2025c) at 1.3B and 14B and evaluate VBench (Huang et al., 2024) on 944 Self-Forcing prompts (Huang et al., 2025). We compare with DMD2 (Yin et al., 2024a), rCM (Zheng et al., 2026), AnyFlow (Gu et al., 2026), and PDD (Shaul et al., 2026). Compared with AnyFlow, our students have higher Dynamic Degree and Motion Smoothness at both scales, alongside competitive Quality but lower Semantic scores (Table 3). Selected outputs show less pronounced saturation than the baselines (Figure 14 in the appendix).

Joint audio–video generation. Joint generation adds the challenge of fitting both visual and acoustic dynamics within a shared inference budget. MiniMax H3 (MiniMax AI, 2026) jointly predicts video and audio. We distill both modalities with equal imitation weights and compare with LightX2V DMD releases (LightX2V and ModelTC, 2026) on 100 matched pairs per budget. The 4-NFE and 8-NFE evaluations use separate sets and model-specific schedules, not a controlled NFE ablation. Our students improve PAM audio clarity (Deshmukh et al., 2024) and Audiobox production quality (Tjandra et al., 2025), with small gains in temporal coherence and visual naturalness (Table 4). Paired intervals support the audio gains but include zero for visual differences. Figure 15 in the appendix illustrates the 8-NFE video comparison. Image/video-conditioned 4-NFE examples, protocols, spectrograms, and uncertainty estimates are in Appendices H.6 and H.7.

4 Beyond Self-Distillation

The same velocity-matching objective extends beyond learning from the student's own pretrained model. We explore two settings that broaden the source of supervision: transferring capabilities from a different model and combining complementary strengths from multiple specialized teachers.

Cross-model supervision. We first consider replacing the student's own teacher with a larger model. Matching output velocities permits such cross-model supervision when prediction representations align. Using imitation alone, we distill FLUX.2-klein-4B and Lens-RL-3.8B students from either their own models or FLUX.2-klein-9B. The larger teacher improves most FLUX.2 metrics, but only diversity and style for Lens. Lens is pretrained with long captions, so short-prompt evaluations may not fully reflect its capabilities (Team, 2026).

Multiple experts as teachers. Beyond transferring from a single teacher, the same objective can consolidate supervision from multiple experts into one student. We distill four reward-specialized teachers for CLIPScore, GenEval, OCR, and PickScore into one 4-NFE SD3.5-Medium student using imitation alone. Routing training prompts to the corresponding experts improves all six reported metrics over self-distillation, including HPSv2 and ImageReward, without surpassing every specialist. Inference uses only the student, with no expert routing. Quantitative results and training details for both experiments are in Appendix F.

5 Ablation Studies

We ablate the design of the Pivot proxy and the effect of teacher repulsion on Qwen-Image and Wan2.1-1.3B. For the Pivot proxy, we examine where to extract student features, how much network

Table 5. Qwen-Image Pivot design at 4 NFE.

Initialization	Overall	Text	Div.
No correction	0.535	0.937	0.127
Pretrained	0.538	0.938	**0.138**
Random	**0.541**	**0.944**	0.135

K	M	Overall	Text	Div.
1	3	0.540	0.943	0.130
4	3	0.541	0.944	**0.135**
1	7	0.534	0.945	0.108
8	7	**0.547**	**0.967**	0.132

Basis	P	Overall	Text	Div.
Polynomial	4	0.538	0.938	0.138
Spectral	4	0.538	0.943	0.129
Collocation	4	0.533	0.928	0.143
Piecewise	4	0.540	0.944	**0.145**
Piecewise	32	**0.543**	**0.962**	0.132

Table 6. Wan2.1-1.3B Pivot design at 4 NFE.

Initialization	Quality	Semantic	Overall
No correction	84.83	76.16	83.09
Pretrained	**85.61**	78.29	**84.14**
Random	85.30	**79.28**	84.10

K	M	Quality	Semantic	Overall
1	3	84.80	77.34	83.31
4	3	85.30	**79.28**	**84.10**
8	3	**85.83**	76.53	83.97

Basis	P	Quality	Semantic	Overall
Polynomial	4	**85.61**	78.29	**84.14**
Spectral	4	84.91	77.76	83.48
Collocation	4	84.79	**79.73**	83.78
Piecewise	4	85.20	78.56	83.87
Piecewise	32	84.59	78.31	83.33

Table 7. Qwen-Image repulsion at teacher guidance 5.

w	λ	Overall	Text	Div.
–	0	0.539	0.943	**0.142**
0	0.1	0.539	**0.960**	0.123
0	0.2	**0.542**	0.958	0.139
1	0.1	0.528	0.899	0.138
1	0.2	0.538	0.938	0.138

Table 8. Wan2.1-1.3B repulsion at teacher guidance 5.

w	λ	Quality	Semantic	Overall
–	0	84.97	77.62	83.50
0	0.1	85.61	**78.29**	84.14
0	0.2	**85.83**	77.59	**84.18**
0	0.4	85.24	77.59	83.71
1	0.1	84.05	77.07	82.65
1	0.2	84.66	75.59	82.85
1	0.4	84.92	78.21	83.57

capacity to allocate, and how to parameterize the within-interval velocity residual. Here K indexes feature taps backward from the student end, M counts Pivot blocks, and P denotes basis size. These comparisons assess how the proxy's access to student representations and its modeling capacity affect distillation quality. We also vary the repulsion weight λ to examine the trade-off between matching the reference teacher and moving away from its weaker predictions. Bold marks the best values within each comparison group. Additional prompt-coverage experiments, complete metric breakdowns, and qualitative comparisons are provided in Appendix G.

Pivot design. Tables 5 and 6 separate the availability of a residual correction from its initialization, feature source, capacity, and basis. The Qwen no-correction control still executes Pivot with a zero residual, whereas Wan omits the Pivot forward pass. Both retain the student velocity head. Adding Pivot improves Overall for both models, and random initialization remains effective. At fixed capacity, earlier features improve the $M = 7$ Qwen comparison, while Wan reaches its best Overall at $K = 4$ rather than the earliest tested location. Increasing decoder depth at $K = 1$ lowers Qwen Overall. Basis preferences differ: Piecewise ($P = 32$) leads on Qwen Overall, while Polynomial ($P = 4$) leads on Wan. More capacity is not uniformly better.

Reference and repulsion strength. We fix teacher guidance at 5 and use a pretrained Pivot with $K = 4$ and $M = 3$ and the Polynomial basis with $P = 4$. The reference CFG scale w is 0 for unconditional predictions and 1 for conditional predictions, and λ controls repulsion strength. At $\lambda = 0$, repulsion is disabled and w is inactive. Tables 7 and 8 show that $w = 0$ with $\lambda = 0.2$ gives the strongest Overall on both models. The $w = 1$ reference does not provide the same aggregate gains over the no-repulsion control. Increasing λ to 0.4 at $w = 0$ lowers Wan Overall, and the Qwen settings trade text quality against diversity. Reference choice and strength have distinct effects.

6 Conclusion

Strong few-step diffusion distillation need not rely on auxiliary score models, adversarial training, or specialized inference architectures. SMD builds on simple on-policy velocity regression with a training-only Pivot proxy for local velocity variation and repulsion from weaker teacher predictions. Across diverse generative tasks and models with up to 33B parameters, SMD delivers strong few-step performance while leaving student inference unchanged. At 4 NFE, it outperforms strong few-step baselines and surpasses multi-step teachers on several benchmarks. The same objective supports cross-model and multi-teacher supervision, establishing on-policy regression as a practical foundation for transferring broad generative capabilities into few-step models.

AI USE STATEMENT

Generative AI tools assisted with code implementation and debugging, experimental planning and result interpretation, literature search, figure preparation, and manuscript drafting and revision. Language models were also used to generate or expand training prompts and editing instructions for parts of the experiments. Some evaluations use language or vision–language models as automated judges, as described in the evaluation protocols. These scores are reported as automated assessments rather than human judgments. The authors are responsible for the final text, experimental claims, citations, code, and other artifacts, including those prepared with AI assistance.

ETHICS STATEMENT

Reducing the computational cost of generative models can broaden access to image, video, audio, and 3D content creation, but can also lower barriers to producing deceptive media, impersonation, and other harmful content. Distillation may retain biases, unsafe behaviors, and limitations of the pretrained teachers, and the reported quality benchmarks do not establish safety or fairness. Qualitative results are presented as model-generated outputs, not records of real events. Third-party models, datasets, and reference media remain subject to their applicable licenses, privacy obligations, and intellectual-property rights. Any release or deployment should respect these restrictions and retain appropriate safeguards and disclosure of generated content.

REPRODUCIBILITY STATEMENT

Section 2 specifies the training objectives and path construction. The training architecture and complete training and inference algorithms are provided in Appendix C, with additional mathematical details in Appendix D.1. Appendix H documents training data, model configurations, optimization settings, checkpoint iterations, and evaluation procedures for the few-step experiments. Section 5 describes the controlled comparisons used to examine the design choices. The H3 evaluation protocol and paired uncertainty estimates are detailed in Appendix H.7. Results quoted from prior work are distinguished from evaluations of released checkpoints, and selected qualitative examples are distinguished from aggregate benchmark results.

REFERENCES

Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024.

Jingjing Chang, Yixiao Fang, Peng Xing, Shuhan Wu, Wei Cheng, Rui Wang, Xianfang Zeng, Gang Yu, and Hai-Bao Chen. Oneig-bench: Omni-dimensional nuanced evaluation for image generation. *arXiv preprint arXiv:2506.07977*, 2025.

Hansheng Chen, Kai Zhang, Hao Tan, Leonidas Guibas, Gordon Wetzstein, and Sai Bi. pi-flow: Policy-based few-step generation via imitation distillation. In *ICLR*, 2026.

Junsong Chen, Shuchen Xue, Yuyang Zhao, Jincheng Yu, Sayak Paul, Junyu Chen, Han Cai, Song Han, and Enze Xie. Sana-sprint: One-step diffusion with continuous-time consistency distillation. *arXiv preprint arXiv:2503.09641*, 2025.

Zhenglin Cheng, Peng Sun, Jianguo Li, and Tao Lin. Twinflow: Realizing one-step generation on large models with self-adversarial flows. In *ICLR*, 2026.

Soham Deshmukh, Dareen Alharthi, Benjamin Elizalde, Hannes Gamper, Mahmoud Al Ismail, Rita Singh, Bhiksha Raj, and Huaming Wang. PAM: Prompting audio-language models for audio quality assessment. *arXiv preprint arXiv:2402.00282*, 2024.

Xingtong Ge, Xin Zhang, Tongda Xu, Yi Zhang, Xinjie Zhang, Yan Wang, and Jun Zhang. Senseflow: Scaling distribution matching for flow-based text-to-image distillation. In *ICLR*, 2026.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. In *NeurIPS*, 2025.

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.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS Workshop*, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. In *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 *CVPR*, 2024.

Hunyuan3D Team, Shuhui Yang, Mingxin Yang, Yifei Feng, Xin Huang, Sheng Zhang, Zebin He, Di Luo, Haolin Liu, Yunfei Zhao, et al. Hunyuan3d 2.1: From images to high-fidelity 3d assets with production-ready pbr material. *arXiv preprint arXiv:2506.15442*, 2025.

Junha Hyung, Kinam Kim, Susung Hong, Min-Jung Kim, and Jaegul Choo. Spatiotemporal skip guidance for enhanced video diffusion sampling. *arXiv preprint arXiv:2411.18664*, 2024. URL https://arxiv.org/abs/2411.18664.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *NeurIPS*, 2022.

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 *ICLR*, 2024.

Sangyun Lee, Beomsu Kim, and Jong Chul Ye. Minimizing trajectory curvature of ode-based generative models. In *International Conference on Machine Learning*, pp. 18957–18973. PMLR, 2023.

LightX2V and ModelTC. Minimax-H3-Turbo. https://github.com/ModelTC/Minimax-H3-Turbo, 2026.

LightX2V Team. Qwen-image-edit-2511-lightning. https://huggingface.co/lightx2v/Qwen-Image-Edit-2511-Lightning, 2026.

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

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. In *NeurIPS*, 2025a.

Shiyu Liu, Yucheng Han, Peng Xing, Fukun Yin, Rui Wang, Wei Cheng, Jiaqi Liao, et al. Step1x-edit: A practical framework for general image editing. *arXiv preprint arXiv:2504.17761*, 2025b.

Tao Liu, Hao Yan, Mengting Chen, Taihang Hu, Zhengrong Yue, Zihao Pan, Jinsong Lan, Xiaoyong Zhu, Ming-Ming Cheng, Bo Zheng, and Yaxing Wang. Continuous-time distribution matching for few-step diffusion distillation. *arXiv preprint arXiv:2605.06376*, 2026.

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

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

ModelTC Team. Qwen-image-lightning. https://github.com/ModelTC/Qwen-Image-Lightning, 2025.

Weili Nie, Julius Berner, Nanye Ma, Chao Liu, Saining Xie, and Arash Vahdat. Transition matching distillation for fast video generation, 2026. URL https://arxiv.org/abs/2601.09881.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

J. O. Ramsay and B. W. Silverman. *Functional Data Analysis*. Springer, 2 edition, 2005. doi: 10.1007/b98888. URL https://link.springer.com/book/10.1007/b98888.

Yuxi Ren, Xin Xia, Yanzuo Lu, Jiacheng Zhang, Jie Wu, Pan Xie, XING WANG, and Xuefeng Xiao. Hyper-SD: Trajectory segmented consistency model for efficient image synthesis. In *NeurIPS*, 2024.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *ICLR*, 2022.

Axel Sauer, Frederic Boesel, Tim Dockhorn, Andreas Blattmann, Patrick Esser, and Robin Rombach. Fast high-resolution image synthesis with latent adversarial diffusion distillation. In *SIGGRAPH Asia*, 2024a.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *ECCV*, 2024b.

Neta Shaul, Chao Liu, Arash Vahdat, and Julius Berner. Parallel decoding distillation for fast image and video generation. *arXiv preprint arXiv:2607.26004*, 2026.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021a.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021b.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *ICML*, 2023.

Lens Team. Lens: Rethinking training efficiency for foundational text-to-image models. *arXiv preprint arXiv:2605.21573*, 2026.

Qwen-Image Team. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025a.

Tencent Hunyuan3D Team. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. *arXiv preprint arXiv:2501.12202*, 2025b.

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

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

Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023.

Jianfeng Xiang, Xiaoxue Chen, Sicheng Xu, Ruicheng Wang, Zelong Lv, Yu Deng, Hongyuan Zhu, Yue Dong, Hao Zhao, Nicholas Jing Yuan, and Jiaolong Yang. Native and compact structured latents for 3d generation. In *CVPR*, 2026.

Yang Ye, Xianyi He, Zongjian Li, Bin Lin, Shenghai Yuan, Zhiyuan Yan, Bohan Hou, and Li Yuan. Imgedit: A unified image editing dataset and benchmark. *arXiv preprint arXiv:2505.20275*, 2025.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T. Freeman. Improved distribution matching distillation for fast image synthesis. In *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 *CVPR*, 2024b.

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

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 *ICLR*, 2026.

Appendix Contents

A Related Work **15**

B Preliminaries **15**

C Training Architecture and Algorithms **16**

C.1 Shared Computation 16

C.2 Training and Inference Algorithms 17

D Supplementary Analysis and Mathematical Details **17**

D.1 Residual Parameterizations 17

D.2 Block-Swap Study 21

E Additional Few-step Results **21**

E.1 Image Generation 21

E.2 Image Editing 23

E.3 3D Shape and Texture 24

E.4 Video Generation 24

E.5 Joint Audio–Video Generation 25

F Additional Cross-Model and Multi-Teacher Results **25**

F.1 Cross-Model Distillation 25

F.2 Multiple Experts as Teachers 27

G Additional Ablation Results **29**

G.1 Prompt Coverage 29

G.2 Pivot Design 32

G.3 Teacher Repulsion 32

H Few-step Implementation and Evaluation Details **34**

H.1 Image Generation 34

H.2 Image Editing 36

H.3 3D Shape and Texture 36

H.4 Video Generation 37

H.5 Detailed VBench Results 37

H.6 Joint Audio–Video Generation 37

H.7 H3 Evaluation and Additional Results 39

A Related Work

Few-step generation reduces the sequential sampling cost of diffusion and flow models. The main approaches differ in what the student learns to reproduce: finite-time transport along a teacher trajectory, a generated distribution, or local teacher predictions along student rollouts. These choices can be combined, so the categories below describe the source of supervision rather than mutually exclusive model families.

Trajectory and flow-map formulations. Progressive distillation (Salimans & Ho, 2022) repeatedly compresses a deterministic teacher sampler by training a student to replace pairs of sampling steps. Consistency models (Song et al., 2023) learn predictions that agree across states on the same probability-flow ODE trajectory, enabling both one-step and multistep generation. Consistency trajectory models (Kim et al., 2024) extend this view to transport between arbitrary initial and final times. MeanFlow (Geng et al., 2025) instead models the interval-averaged velocity and derives an identity relating it to the instantaneous velocity, supporting one-step training without a pretrained teacher. These formulations give the network a finite-time prediction target rather than simply evaluating a local velocity field fewer times. Our student also predicts an interval-level update, but is trained against local teacher velocities at states constructed along its own transition. Pivot changes the within-interval proxy used for supervision while preserving the endpoint of that coarse update.

Distribution matching and adversarial distillation. Distribution-level methods supervise generated samples through learned distributional feedback. DMD (Yin et al., 2024b) uses the difference between teacher and learned student-distribution scores to update a few-step generator. DMD2 (Yin et al., 2024a) improves score-estimation stability with two-time-scale updates and incorporates an adversarial objective using real data. Adversarial Diffusion Distillation (Sauer et al., 2024b) combines teacher supervision with a discriminator, while Latent Adversarial Diffusion Distillation (Sauer et al., 2024a) uses features from pretrained latent diffusion models for discrimination. These approaches allow distributional criteria to supplement or replace samplewise regression. Our objective instead compares student and teacher velocities directly, without online estimation of the student score or a discriminator. The distinction concerns the training signal and auxiliary models, not an assumption that every distribution-matching method has the same quality or diversity behavior.

On-policy regression and within-interval prediction. Pi-Flow (Chen et al., 2026) and PDD (Shaul et al., 2026) are closely related regression-based approaches. Pi-Flow predicts a network-free policy that supplies time-varying velocities for integration over substeps within a network evaluation. PDD predicts multiple denoising updates in parallel, supporting different sampling budgets through its decoding formulation. Both motivate retaining more information about teacher dynamics than a single instantaneous prediction. SMD places this additional capacity in a training-only residual branch. Its basis functions integrate to zero, so the branch can modify local velocities and query states without changing the student's coarse endpoint. Sampling therefore uses the original student output head directly, without evaluating Pivot or integrating a predicted substep policy.

Guidance and reference predictions. Classifier-free guidance (Ho & Salimans, 2021) combines conditional and unconditional predictions to control the fidelity–coverage trade-off. Spatiotemporal Skip Guidance (Hyung et al., 2024) constructs a weaker prediction through self-perturbation of the same model for use during sampling. Our reference constructions are related to these guidance mechanisms, but their role is different: a reference prediction supplies the repulsive term during distillation. The teacher target remains the prediction used for imitation, and the reference contribution is controlled separately by its weight. No reference-model evaluation is added to the student sampler.

B Preliminaries

Flow-based generative models. Modern diffusion and flow models can be formulated in terms of a learned velocity field that transports a simple noise distribution to the data distribution (Lipman et al., 2023; Liu et al., 2023). Let $x_0 \sim p_{\text{data}}(x \mid c)$ be a data sample under condition c, and let $x_1 \sim \mathcal{N}(0, I)$ be Gaussian noise. A common linear probability path is

$$x_t = (1-t)x_0 + tx_1, \qquad t \in [0,1], \tag{11}$$

where $t = 0$ is the data endpoint and $t = 1$ is the noise endpoint. The conditional velocity along this path is $\dot{x}_t = x_1 - x_0$. Flow matching trains a velocity model to predict this quantity from the interpolated state; we write its prediction as $v_T(x_t, t, c)$ when it serves as the teacher,

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}_{x_0, x_1, t}\left[\|v_T(x_t, t, c) - (x_1 - x_0)\|_2^2\right]. \tag{12}$$

At the population optimum, the learned field approaches the marginal velocity $v^*(x, t, c) = \mathbb{E}[x_1 - x_0 \mid x_t = x, c]$. This velocity defines the generative ODE

$$\frac{dx_t}{dt} = v_T(x_t, t, c). \tag{13}$$

Generation starts from $x_1 \sim \mathcal{N}(0, I)$ and integrates the ODE from $t = 1$ to $t = 0$. Each model evaluation provides only the local velocity at the current state and time. A complete sample is obtained by integrating these predictions over the full time interval. The state x_t may be an image or a learned latent, and c may contain text, images, or other conditioning signals (Rombach et al., 2022; Peebles & Xie, 2023). Some pretrained models natively predict noise or a clean sample rather than velocity. We convert these outputs to velocity under the model's original parameterization. As in Section 2, v_T is frozen during distillation, v_θ denotes the student, and ϕ denotes the Pivot parameters.

Conditional guidance. Classifier-free guidance combines predictions with and without the condition (Ho & Salimans, 2021). In velocity form, a guidance scale w gives

$$v_T^{(w)}(x_t, t, c) = v_T(x_t, t, \varnothing) + w\left[v_T(x_t, t, c) - v_T(x_t, t, \varnothing)\right]. \tag{14}$$

The conditional prediction is recovered at $w = 1$, while larger values generally strengthen conditioning at a possible cost to diversity. The weaker prediction need not come from an unconditional branch. Recent guidance methods may instead use a weaker model, a perturbed forward pass, or a reduced set of conditions. In each case, guidance modifies the local velocity used by the sampler without changing the underlying integration procedure.

Sampling schedules. Although the learned velocity field is continuous in time, inference evaluates it on a finite decreasing grid

$$1 = t_0 > t_1 > \cdots > t_N = 0. \tag{15}$$

The placement of these time points determines which regions of the transport receive more model evaluations. Some pretrained models additionally warp the nominal time before passing it to the network. One common rational shift is

$$S_\gamma(t) = \frac{\gamma t}{1 + (\gamma - 1)t}, \qquad \gamma > 0. \tag{16}$$

We absorb this transformation into the sampling grid and use t for the resulting model time below. Given $\Delta_i = t_i - t_{i+1} > 0$, a first-order update takes the form

$$x_{i+1} = x_i - \Delta_i v_T(x_i, t_i, c). \tag{17}$$

Higher-order solvers evaluate the field at additional states within an interval to reduce discretization error (Song et al., 2021a; Karras et al., 2022). Each network call is counted as one function evaluation, or NFE, and constitutes the main inference cost. Standard samplers use many evaluations to follow the learned field closely. Few-step sampling retains only a small number of coarse transitions, making the accuracy of each transition increasingly important.

C Training Architecture and Algorithms

C.1 Shared Computation

Pivot takes student features at position K, counted backward from the end, and processes them with M Transformer blocks. Thus $K = 1$ denotes the last feature-tap position, while M independently specifies branch depth. For pretrained initialization, the branch copies the final M student blocks. The feature computation preceding the tap is shared with the student. Its output projection predicts all P coefficient vectors in a single forward pass and is zero-initialized, so training starts from the baseline transition. The choice of a late branch is also informed by the block-swap study in Appendix D.2.

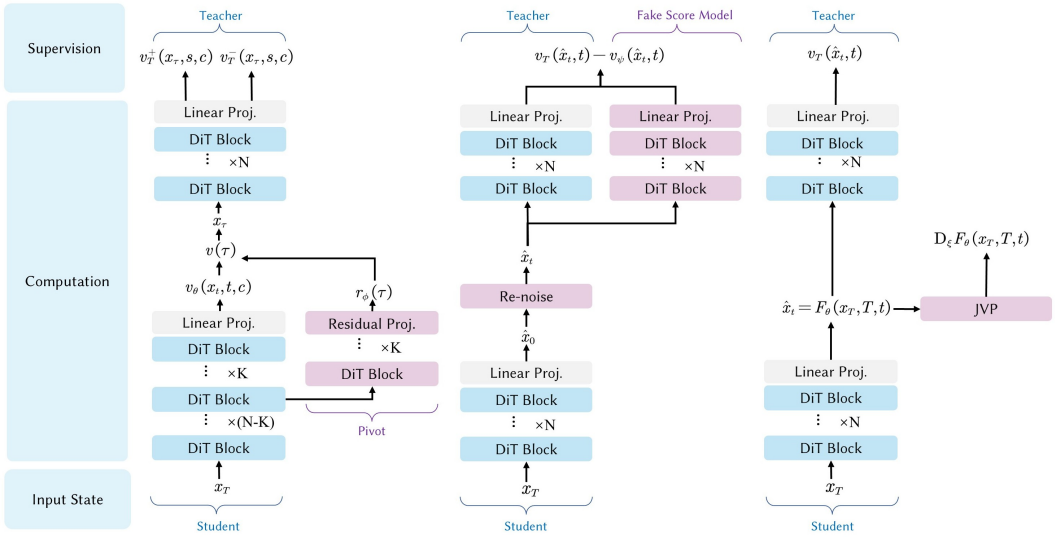


Figure 6. **Training architectures for few-step distillation.** (a) The student and Pivot share backbone features to predict a coarse velocity and residual coefficients. The residual defines the proxy states and local velocities used for teacher supervision; Pivot is omitted at inference. For reference, (b) shows a distribution-matching formulation with re-noising and an auxiliary fake-score model, and (c) shows a flow-map formulation that obtains a local velocity through a Jacobian–vector product (JVP). The latter is illustrated with source state x_T.

Training follows the student's few-step sampling schedule. At each transition, the student predicts a coarse velocity and Pivot predicts the residual coefficients using shared backbone features, as illustrated in Figure 6(a). We sample $\tau \sim \mathcal{U}(0,1)$, construct the corresponding proxy state and local velocity, and evaluate the target and reference teacher predictions at that state. The resulting objective in Eq. (10) jointly trains the student and Pivot.

C.2 Training and Inference Algorithms

Algorithms 2 and 3 specify the training loss and student sampler. Here $\mathrm{StudentPivot}$ denotes the shared forward computation that produces the student velocity and Pivot coefficients, and $\mathrm{stopgrad}$ denotes gradient detachment. The loss is averaged over N sampling intervals, with batch averaging and optimizer updates handled outside the rollout.

D Supplementary Analysis and Mathematical Details

D.1 Residual Parameterizations

From a velocity residual to a path correction. Recall that $\tau \in [0,1]$ measures progress through a segment of duration Δ. Pivot predicts P coefficient vectors a_m, each with the same shape as the student velocity, and holds them fixed within the segment. Scalar basis functions determine how their contributions vary with τ:

$$r_\phi(\tau) = \sum_{m=1}^{P} a_m b_m(\tau), \qquad q_\phi(\tau) = \sum_{m=1}^{P} a_m I_m(\tau), \qquad I_m(\tau) = \int_0^\tau b_m(u)\,du. \tag{18}$$

Here r_ϕ modifies the local velocity, whereas its integral q_ϕ determines the accumulated correction to the straight-line path:

$$v(\tau) = v_\theta + r_\phi(\tau), \qquad x_\tau = x_t - \Delta\big[\tau v_\theta + q_\phi(\tau)\big]. \tag{19}$$

Algorithm 2: On-policy distillation with a proxy path

Input: Student parameters θ, Pivot parameters ϕ, frozen teacher prediction rules v_T and $\widetilde{v}_T$; condition c.
Input: Time grid $T = t_0 > t_1 > \cdots > t_N = 0$, zero-integral bases $\{b_m\}_{m=1}^{P}$, repulsion weight λ.
Output: Rollout loss $\mathcal{L}$ for updating θ and ϕ.

Sample $x \sim \mathcal{N}(0, I)$;
$\mathcal{L} \leftarrow 0$;
for $i = 0, \ldots, N-1$ **do**
  $x \leftarrow \mathrm{stopgrad}(x)$;
  $\Delta \leftarrow t_i - t_{i+1}$;
  $v_\theta, \{a_m\}_{m=1}^{P} \leftarrow \mathrm{StudentPivot}(x, t_i, c; \theta, \phi)$;
  Sample $\tau \sim \mathcal{U}(0,1)$; $s \leftarrow t_i - \tau\Delta$;
  $r_\phi(\tau) \leftarrow \sum_{m=1}^{P} a_m b_m(\tau)$;
  $v(\tau) \leftarrow v_\theta + r_\phi(\tau)$;
  $x_\tau \leftarrow x - \Delta\left[\tau v_\theta + \sum_{m=1}^{P} a_m \int_0^\tau b_m(r)\, dr\right]$;
  $v_T \leftarrow \mathrm{stopgrad}\big(v_T(x_\tau, s, c)\big)$;
  $\widetilde{v}_T \leftarrow \mathrm{stopgrad}\big(\widetilde{v}_T(x_\tau, s, c)\big)$;
  $\mathcal{L} \leftarrow \mathcal{L} + \|v(\tau) - v_T\|_2 - \lambda\|v(\tau) - \widetilde{v}_T\|_2$;
  $x \leftarrow x - \Delta v_\theta$; // Continue the student rollout
return $\mathcal{L}/N$;

Algorithm 3: Few-step student inference

Input: Trained student v_θ, condition c, time grid $T = t_0 > \cdots > t_N = 0$.
Output: Generated sample $\hat{x}_0$.

Sample $x \sim \mathcal{N}(0, I)$;
for $i = 0, \ldots, N-1$ **do**
  $\Delta \leftarrow t_i - t_{i+1}$;
  $x \leftarrow x - \Delta v_\theta(x, t_i, c)$;
return $\hat{x}_0 = x$;

Thus the actual state displacement relative to the straight line is $-\Delta q_\phi(\tau)$. Requiring every b_m to have zero integral gives

$$\int_0^1 b_m(\tau)\, d\tau = 0 \quad \Longrightarrow \quad q_\phi(0) = q_\phi(1) = 0 \quad \Longrightarrow \quad x_{\tau=1} = x_t - \Delta v_\theta. \tag{20}$$

The residual can therefore change intermediate states without moving either endpoint for a fixed v_θ. It need not vanish at the endpoints: the constraint concerns its integral, not its endpoint values. Figure 7 compares how the four constructions fit the same challenging path correction as the coefficient budget increases. The controlled fitting protocol is described below.

Polynomial path: construct the path first. A direct way to preserve endpoints is to start with functions that vanish at both ends:

$$B_m(\tau) = \tau(1-\tau)(2\tau-1)^{m-1}, \qquad b_m(\tau) = \frac{dB_m(\tau)}{d\tau}, \quad 1 \le m \le P. \tag{21}$$

The factor $\tau(1-\tau)$ anchors the path correction to zero at both endpoints; successive powers of $2\tau - 1$ add polynomial variation within the interval. Since $B_m(0) = 0$, the integrated basis is simply $I_m = B_m$ and $q_\phi = \sum_m a_m B_m$. Differentiating this path gives r_ϕ, with zero integral because $B_m(1) - B_m(0) = 0$. For example, the first mode has $B_1(\tau) = \tau(1-\tau)$ and $b_1(\tau) = 1 - 2\tau$: its velocity contribution changes sign halfway through the segment, producing a single path excursion that returns to zero. With P coefficients, this construction spans P independent polynomial modes.

Spectral basis: combine oscillatory velocity modes. Instead of constructing the path first, we can directly choose velocities whose positive and negative contributions cancel:

$$b_m(\tau) = \cos(m\pi\tau), \qquad I_m(\tau) = \frac{\sin(m\pi\tau)}{m\pi}, \quad 1 \le m \le P. \tag{22}$$

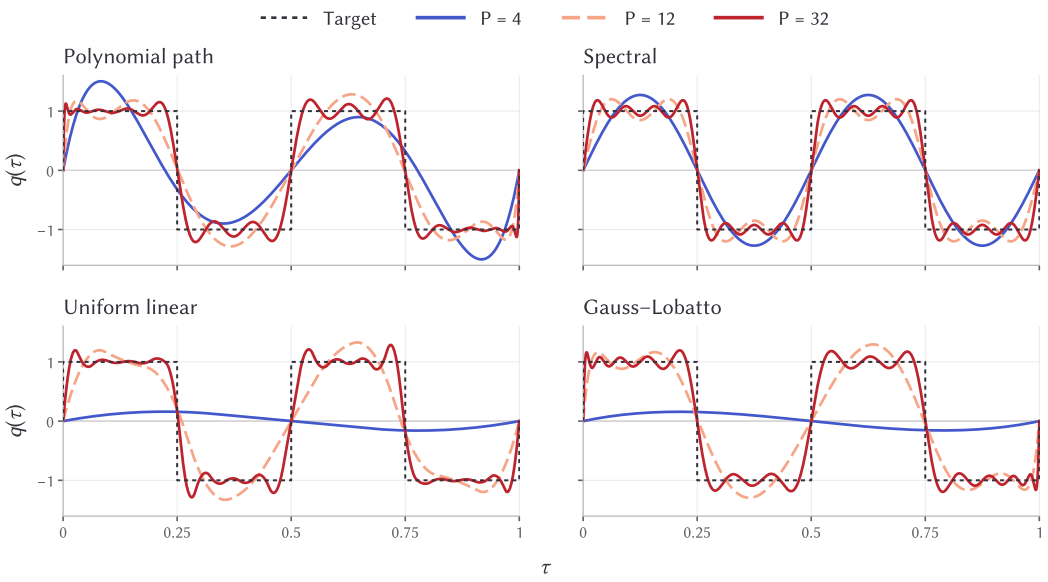


Figure 7. **Endpoint-preserving path approximation.** Unregularized least-squares fits to two full periods of the square wave in Eq. (27) (dotted), with $P \in \{4, 12, 32\}$ coefficients and shared axis scales. All curves have zero endpoint corrections. Polynomial path and Spectral have P independent modes, whereas the nodal constructions have $P - 1$; Polynomial path and Collocation coincide at matched independent dimension. The figure labels Uniform linear and Gauss–Lobatto refer to Piecewise and Collocation, respectively. This illustrates approximation properties, not a ranking of distillation performance.

Each $b_m(\tau)$ integrates to zero on $[0, 1]$. Low indices describe broad changes across the segment, while higher indices allow more rapid oscillations. Integration reduces the amplitude of mode m by $1/(m\pi)$, so equal velocity coefficients do not produce equal path-displacement amplitudes. The constant mode $m = 0$ is excluded because it would change the coarse endpoint. The remaining P cosine modes are mutually orthogonal and independent.

Nodal constructions: interpolate, then subtract the mean. The remaining two designs predict values a_m at $P \geq 2$ nodes τ_m. Let $\ell_m(\tau)$ be interpolation functions satisfying $\ell_m(\tau_k) = \delta_{mk}$. The coefficients are the nodal values of an *uncentered* velocity interpolant $\widetilde{r}_\phi(\tau) = \sum_m a_m \ell_m(\tau)$. To remove its interval mean, define

$$w_m = \int_0^1 \ell_m(\tau)\, d\tau, \qquad b_m(\tau) = \ell_m(\tau) - w_m. \tag{23}$$

The resulting residual is $r_\phi(\tau) = \widetilde{r}_\phi(\tau) - \mu_\phi$, where $\mu_\phi = \sum_m w_m a_m$ is the mean of $\widetilde{r}_\phi$ over the unit interval. Every b_m integrates to zero because $\int_0^1 (\ell_m - w_m) = w_m - w_m = 0$. Its path contribution is

$$I_m(\tau) = \int_0^\tau \ell_m(u)\, du - \tau w_m. \tag{24}$$

At node τ_m, the final residual equals $a_m - \mu_\phi$, rather than a_m. This centering operation makes arbitrary predicted nodal values compatible with the endpoint constraint.

Piecewise basis: uniform piecewise-linear interpolation. We place nodes uniformly at $\tau_m = (m - 1)/(P - 1)$ and interpolate adjacent values with straight lines. The interpolation functions ℓ_m are triangular hat functions (half-hats at the two endpoints). Their areas give the exact integration weights:

$$w_1 = w_P = \frac{1}{2(P-1)}, \qquad w_m = \frac{1}{P-1} \quad (2 \leq m \leq P-1). \tag{25}$$

Centering preserves the piecewise-linear shape of the velocity residual, whose integral is a continuously differentiable, piecewise-quadratic path correction. Adding nodes allows finer variation within

the segment. Before centering, changing one nodal value affects only its adjacent intervals; after centering, it also changes the constant mean subtracted over the entire segment. Thus the centered basis itself does not have strictly local support.

Collocation basis: Gauss–Lobatto Lagrange interpolation. Gauss–Lobatto nodes include both endpoints and, for $P > 3$, are spaced nonuniformly with higher density near the ends. We use the global Lagrange interpolation functions

$$\ell_m(\tau) = L_m(\tau) = \prod_{k \neq m} \frac{\tau - \tau_k}{\tau_m - \tau_k}. \tag{26}$$

Here $w_m = \int_0^1 L_m(\tau)\, d\tau$ are the corresponding Gauss–Lobatto quadrature weights on $[0, 1]$. Subtracting w_m as in Eq. (23) produces a zero-mean polynomial velocity residual of degree at most $P - 1$ and a path correction of degree at most P. Unlike piecewise-linear interpolation, each nodal value influences a polynomial across the whole interval.

How many independent modes? Both nodal constructions reproduce constant functions, so $\sum_m \ell_m(\tau) = 1$ and $\sum_m w_m = 1$. Consequently, $\sum_m b_m(\tau) = 0$. Adding the same vector to all a_m leaves r_ϕ unchanged, so one coefficient direction is redundant. They therefore have $P - 1$ independent modes per velocity coordinate, compared with P for Polynomial path and Spectral. Table 9 summarizes the resulting function spaces.

A controlled path-fitting example. To expose approximation behavior, we fit a single scalar coordinate of q_ϕ while holding the coarse path fixed. The target contains two full periods of a square wave, alternating between positive and negative half-cycles:

$$q_*(\tau) = \begin{cases} 1, & \tau \in (0, \frac{1}{4}) \cup (\frac{1}{2}, \frac{3}{4}), \\ -1, & \tau \in (\frac{1}{4}, \frac{1}{2}) \cup (\frac{3}{4}, 1), \\ 0, & \tau \in \{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}. \end{cases} \tag{27}$$

This discontinuous target is an approximation stress test, not an admissible continuous proxy path. The values at the jumps are set to zero, including the two endpoints; these isolated values do not affect the integrated fitting objective. It makes transition width, overshoot, and oscillations on the flat regions easy to compare. For each family and $P \in \{4, 12, 32\}$, we solve

$$\min_{a_1, \ldots, a_P} \int_0^1 \left[\sum_{m=1}^{P} a_m I_m(\tau) - q_*(\tau) \right]^2 d\tau. \tag{28}$$

This is unregularized least squares in path space, distinct from the teacher-velocity objective used for distillation. We use Gauss quadrature split at the target jumps and interpolation knots, solve with SVD, and verify RMSE on an independent 160k-point grid. Polynomial path is evaluated in an equivalent Jacobi polynomial basis to avoid high-order monomial ill-conditioning; Collocation uses Legendre–Gauss–Lobatto nodes and stable Legendre coordinates. These coordinate changes preserve the fitted function spaces.

At low P, the fits replace the jumps with broad transitions. Increasing P sharpens the transitions, while overshoot and oscillations remain visible. The global polynomial and spectral fits distribute oscillations across the interval; the dense piecewise-quadratic Piecewise fit concentrates them more closely around the jumps in this example. Because all fitted paths are continuous, none can reproduce the jumps exactly at finite P. This experiment isolates finite-dimensional approximation behavior under a specified path loss; it does not measure optimization or regularization bias, nor establish a ranking during distillation.

The relationship between Polynomial path and Collocation is worth distinguishing from these target-specific observations. With d independent modes, both span all polynomials of degree at most $d + 1$ that vanish at 0 and 1: Polynomial path uses $P = d$, while Collocation uses $P = d + 1$. Their best fits therefore coincide when d is matched, which we also verify numerically. Differences at the same P reflect the extra polynomial mode available to Polynomial path, not different approximation spaces at matched dimension. The two coefficient representations can still differ in conditioning and in their interaction with regularization and network optimization.

Table 9. Endpoint-preserving residual parameterizations with P coefficients. Modes are counted per velocity coordinate; nodal constructions require $P \geq 2$.

Parameterization	**Velocity residual** r_ϕ	**Path correction** q_ϕ	**Modes**
Polynomial path	Polynomial, degree $\leq P$	Polynomial, degree $\leq P+1$	P
Spectral	Cosine modes $1, \ldots, P$	Sine modes scaled by $1/(m\pi)$	P
Piecewise	Continuous, piecewise linear	Continuously differentiable, piecewise quadratic	$P-1$
Collocation	Polynomial, degree $\leq P-1$	Polynomial, degree $\leq P$	$P-1$



Figure 8. **Late blocks shape few-step generation.** The left half replaces one block group in each Base model with its Turbo counterpart, while the right half performs the reverse replacement. Rows correspond to Sana, Krea-2, and SD3.5-Large.

D.2 Block-Swap Study

We study where to attach Pivot using bidirectional block swaps on Base–Turbo pairs of Krea-2, SD3.5-Large, and SANA-Sprint (Chen et al., 2025). We exchange the early, middle, or late third of the blocks and evaluate the same few-step schedule with one image per prompt on 100 HPSv2 prompts (Wu et al., 2023). The Base $\rightarrow$ Turbo columns in Table 10 start from Base and insert the corresponding Turbo blocks; the Turbo $\rightarrow$ Base columns reverse this operation. In all three pairs, inserting Turbo's late blocks into Base gives the highest scores among the tested groups, while replacing Turbo's late blocks with Base blocks gives the lowest. Figure 8 shows the corresponding qualitative comparisons. These observations motivate the branch placement in Section 2.2.

Table 10. Bidirectional block swaps evaluated by mean HPSv2 over the same 100 prompts.

	Base $\rightarrow$ Turbo			Turbo $\rightarrow$ Base		
Model	Early	Mid	Late	Early	Mid	Late
Sana	.138	.231	**.269**	**.297**	.273	.242
Krea-2	.175	.205	**.291**	**.304**	.301	.216
SD3.5-L	.151	.198	**.288**	**.299**	.289	.202

E Additional Few-step Results

This appendix provides full metric breakdowns and qualitative comparisons for Section 3. Implementation and evaluation details are provided in Appendix H.

E.1 Image Generation

Tables 11 and 12 supplement Table 1 with all five OneIG-Bench-EN dimensions (Chang et al., 2025). Figures 9 and 10 show repeated generations for the same prompts, allowing comparison of viewpoint, pose, and composition. Baseline results use released checkpoints, except for PDD, whose Euler-variant scores are quoted from its paper (Shaul et al., 2026).

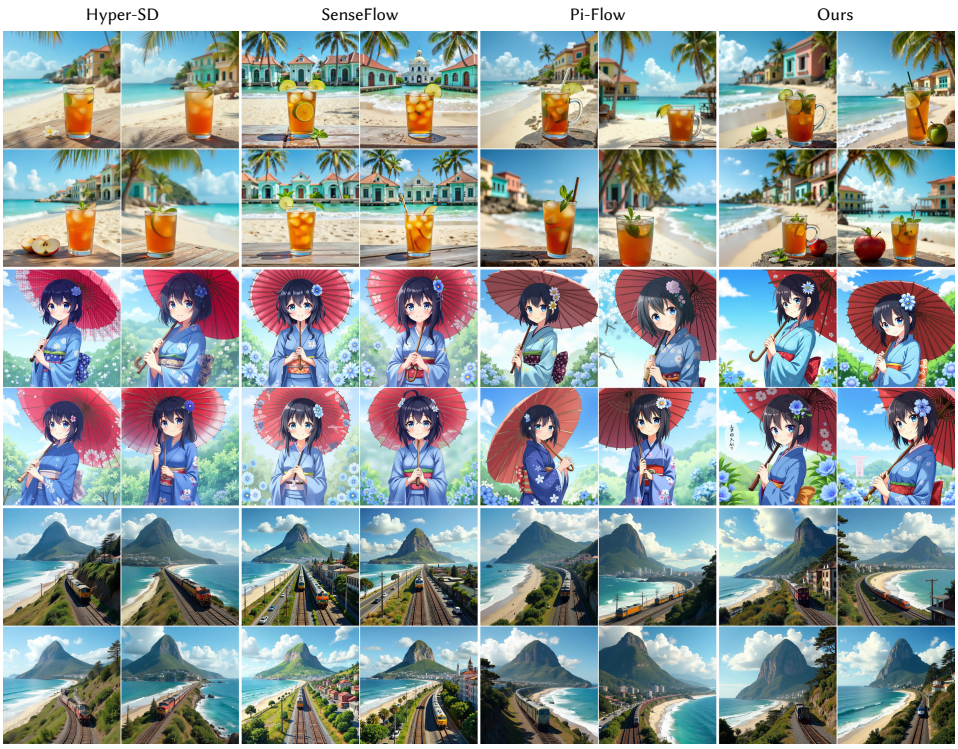


Figure 9. **Image quality and variation with FLUX.1-dev.** Each two-column group shows four 4-NFE samples per prompt for one method, allowing comparison of composition and detail.

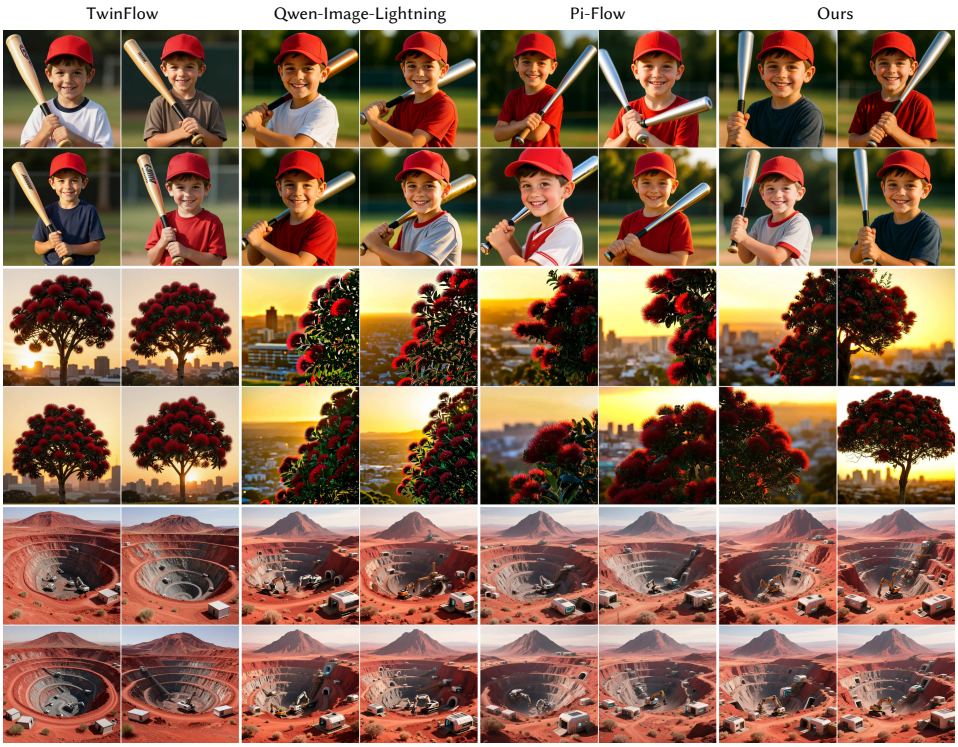


Figure 10. **Image quality and variation with Qwen-Image.** Each two-column group shows four 4-NFE samples per prompt for one method, allowing comparison of pose, viewpoint, and composition.

Table 11. Full OneIG-Bench-EN results for FLUX.1-dev, supplementing Table 1. All scores are higher-is-better. Bold and underline mark the best and second-best 4-NFE results; the full-budget Teacher is excluded from this ranking.

Method	NFE	Overall	Align.	Text	Diversity	Style	Reasoning
Teacher	50	0.430	0.790	0.556	0.239	0.308	0.258
Hyper-SD	4	0.399	0.786	0.423	0.185	<u>0.361</u>	0.239
SenseFlow	4	0.380	0.774	0.388	0.156	0.348	0.234
Pi-Flow	4	<u>0.419</u>	<u>0.799</u>	**0.460**	**0.229**	0.360	<u>0.251</u>
Ours	4	**0.420**	**0.810**	<u>0.455</u>	<u>0.209</u>	**0.370**	**0.256**

Table 12. Full OneIG-Bench-EN results for Qwen-Image, supplementing Table 1. All scores are higher-is-better. Bold and underline mark the best and second-best 4-NFE scores, excluding Teacher; 50×2 counts both CFG branches. † denotes Euler-variant PDD results quoted from its paper (Shaul et al., 2026).

Method	NFE	Overall	Align.	Text	Diversity	Style	Reasoning
Teacher	50×2	0.540	0.880	0.889	0.195	0.428	0.307
Lightning	4	0.527	**0.886**	<u>0.921</u>	0.108	0.411	**0.310**
TwinFlow	4	0.502	0.860	0.872	0.152	0.361	0.263
Pi-Flow	4	0.532	0.878	0.876	<u>0.180</u>	**0.430**	0.296
PDD†	4	<u>0.535</u>	0.879	0.897	**0.192**	0.417	0.292
Ours	4	**0.540**	<u>0.883</u>	**0.944**	0.144	<u>0.429</u>	<u>0.300</u>

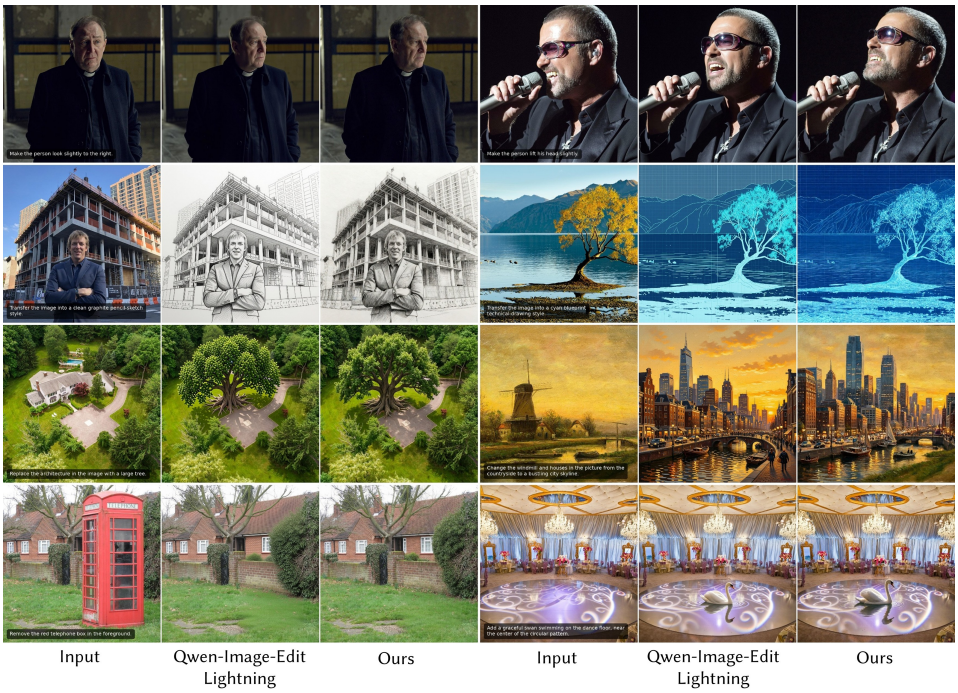


Figure 11. **Instruction-guided image editing.** Each triplet shows the input, Lightning, and ours with Qwen-Image-Edit-2511 at 4 NFE under the same editing instruction.

E.2 Image Editing

Table 13 expands the aggregate results in Table 2 into the individual ImgEdit-Bench operations (Ye et al., 2025) and GEdit-Bench-EN metrics (Liu et al., 2025b). The largest gains over Lightning occur in removal and adjustment, while Lightning scores higher on extraction. Figure 11 shows the input and two student outputs for each instruction.

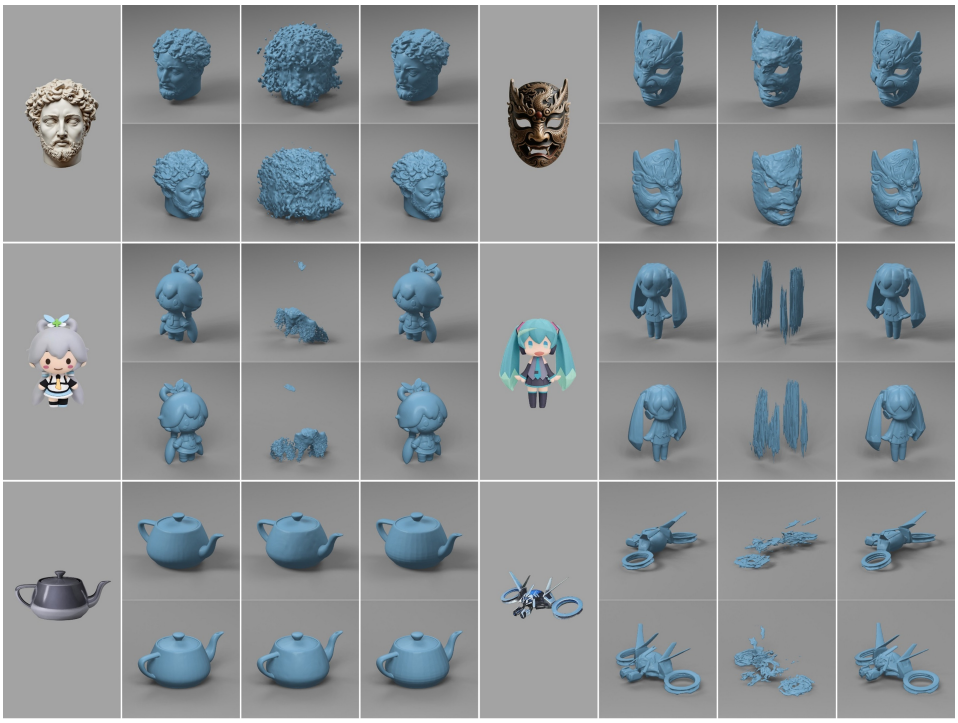


Figure 12. **Few-step image-to-shape generation.** Hunyuan3D 2.1 outputs are compared under the same input images. Each example shows the input image and two rendered views from ours, the reduced-budget teacher, and the full-budget teacher. The student and reduced-budget teacher are evaluated at 4 NFE for shape generation.

Table 13. Full image-editing results for Qwen-Image-Edit-2511. ImgEdit columns report Overall, Adjustment, Style, Background, Extraction, Removal, Replacement, Addition, Hybrid, and Action. GEdit-Bench-EN reports semantic consistency (SC), perceptual quality (PQ), and overall quality (O). All metrics are higher-is-better; bold marks the better 4-NFE student result, including ties. The full-budget Teacher is a reference.

Method	NFE	ImgEdit-Bench										GEdit-Bench-EN		
		Overall	Adj.	Style	BG	Ext.	Rem.	Repl.	Add	Hybr.	Act.	SC	PQ	O
Teacher	40	4.51	4.57	4.89	4.36	4.13	4.46	4.70	4.54	4.16	4.81	8.30	8.20	7.88
Lightning	4	4.48	4.40	**4.76**	**4.39**	**4.21**	4.54	4.72	**4.67**	3.87	**4.80**	8.95	7.74	8.14
Ours	4	**4.51**	**4.58**	4.74	4.37	3.98	**4.74**	**4.76**	**4.67**	**3.95**	**4.80**	**8.96**	**7.79**	**8.19**

E.3 3D Shape and Texture

Figures 12 and 13 provide rendered views, surface visualizations, and detail crops for the qualitative study in Section 3. Each comparison includes the distilled student, direct reduced-budget teacher sampling, and the full-budget teacher. The 4-NFE budget applies to Hunyuan3D's shape stage and separately to each distilled TRELLIS.2 stage, not to its entire asset-generation pipeline.

E.4 Video Generation

Figure 14 supplements Table 3 with ordered frames from the 14B models. The complete 16-dimension VBench results and evaluation protocol are in Appendix H.5. Color adherence measures the specified object colors, not saturation or visual naturalness; selected frame appearance should not be interpreted as a measurement of that metric.

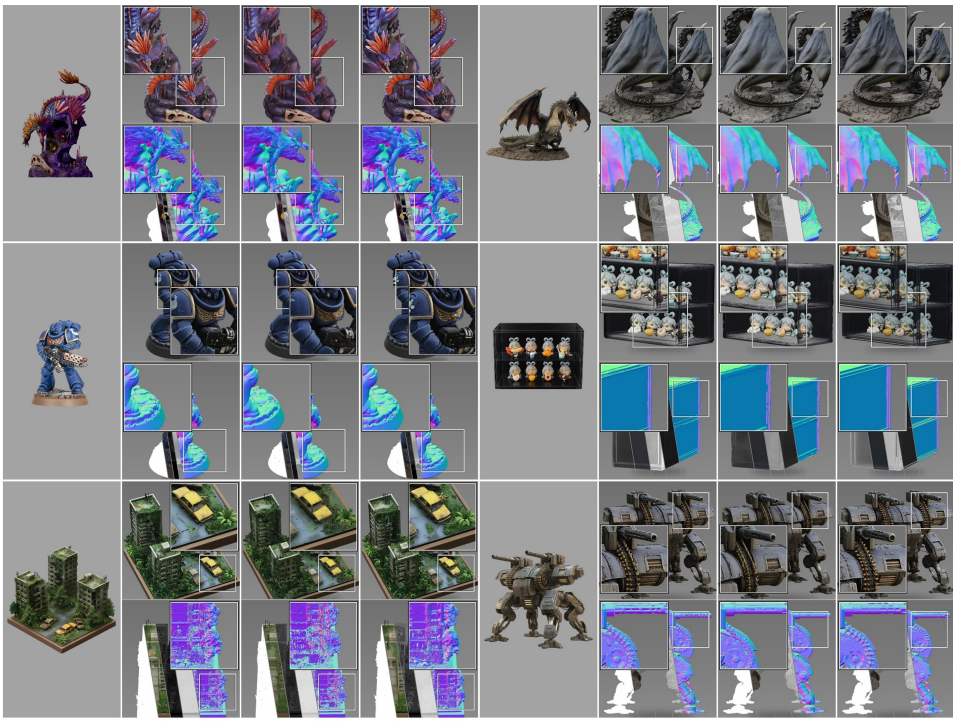


Figure 13. **Few-step shape and texture generation.** TRELLIS.2 renders, detail crops, and surface visualizations compare ours with reduced-budget and full-budget teachers. Only the final high-resolution shape and texture stages are distilled, each to 4 NFE. The preceding sparse-structure and coarse-shape stages remain unchanged.

E.5 Joint Audio–Video Generation

Figure 15 provides 8-NFE examples for the H3 comparison in Table 4, with three chronological frames at matched timestamps for each prompt–seed pair. Appendix H.6 gives the paired confidence intervals, evaluation protocol, and additional 4-NFE examples with audio spectrograms. The 4-NFE and 8-NFE comparisons use different evaluation sets and separately trained models, rather than constituting a controlled ablation of inference budget.

We also explore reference-conditioned audio–video generation at 4 NFE. Training uses instructions paired with reference audio–video clips, without ground-truth target outputs. Figure 16 illustrates character appearance editing, multi-image composition, reference-guided action, and appearance transfer from an image to a video. These selected examples show that the distilled model retains several reference-conditioned capabilities, although fine-grained motion and reference correspondence remain imperfect. We use this setting as a qualitative extension, without an aggregate benchmark or a DMD comparison.

F Additional Cross-Model and Multi-Teacher Results

This appendix reports the full cross-model and multi-teacher comparisons, their training settings, and qualitative examples.

F.1 Cross-Model Distillation

We compare self-distillation with cross-model supervision for two 4-NFE students. For FLUX.2-klein-4B, we use either the same 4B model or FLUX.2-klein-9B as the teacher. For Lens-RL-3.8B (Team, 2026), we compare its own teacher with FLUX.2-klein-9B. This design separates cross-capacity transfer within one model family from cross-architecture transfer between FLUX.2 and Lens.

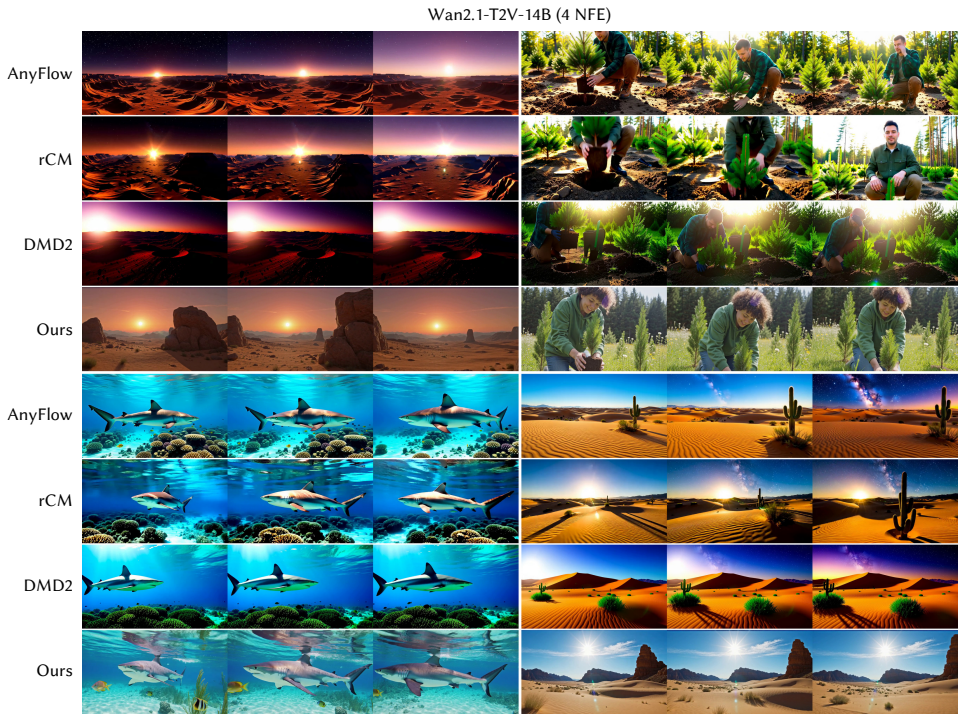


Figure 14. **Video synthesis in four evaluations.** Wan2.1-T2V-14B methods are compared under matched prompts, with three chronological frames per video.

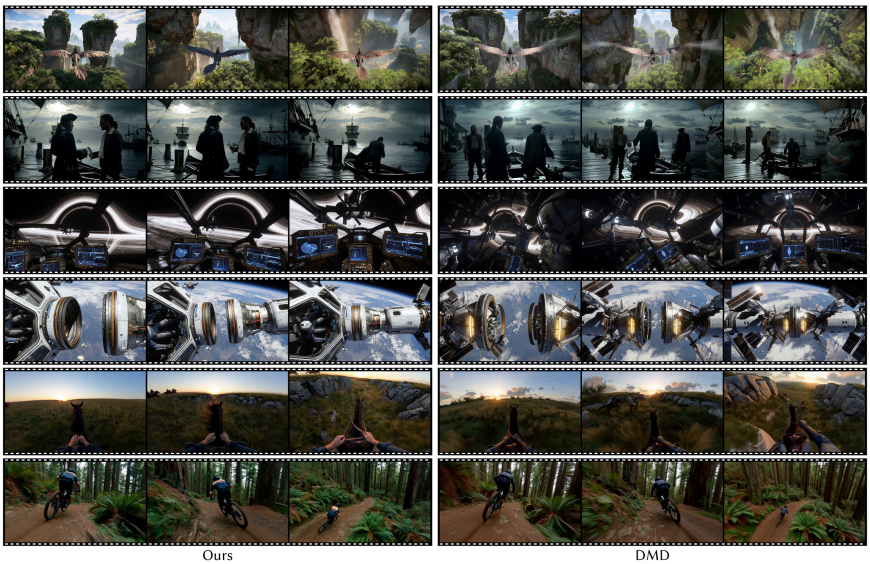


Figure 15. **Video frames from joint audio–video generation.** At 8 NFE, each H3 example shows three chronological frames from ours (left) and DMD (right) at matched timestamps for the same prompt–seed pair.

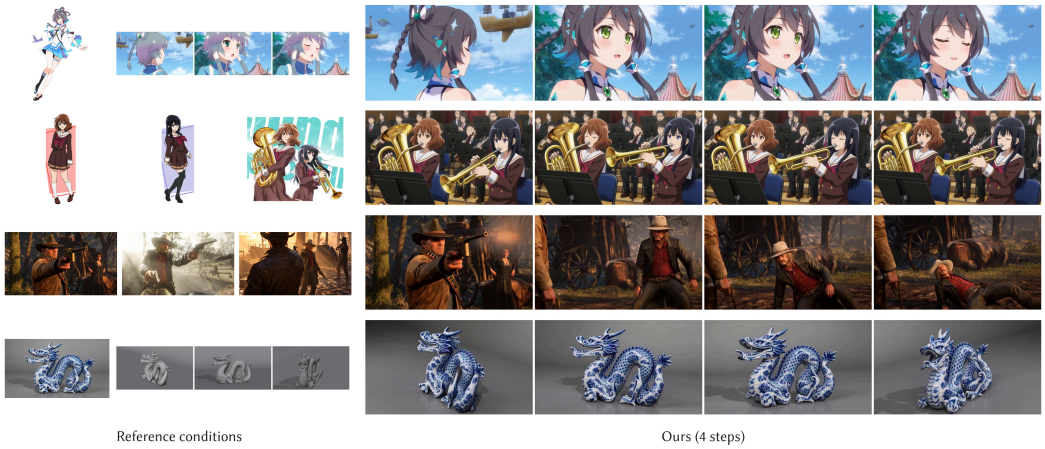


Figure 16. **Reference-conditioned generation in four steps.** H3 input images and video frames appear on the left, with four chronological output frames on the right. Rows show character appearance editing, multi-image composition, reference-guided action, and porcelain appearance transfer from a material image to a gray-model video. All examples use the same 4-NFE student checkpoint.

Table 14. Cross-model distillation to FLUX.2-klein-4B and Lens-RL-3.8B at 4 NFE. Bold marks the better teacher choice for each student and metric.

Student	Teacher	GenEval	DPG	Align.	Text	Div.	Style	Reason.
FLUX.2-klein-4B	FLUX.2-klein-4B	0.789	83.34	0.826	0.663	0.215	0.419	**0.249**
	FLUX.2-klein-9B	**0.791**	**84.26**	**0.843**	**0.744**	**0.224**	**0.441**	0.246
Lens-RL-3.8B	Lens-RL-3.8B	**0.829**	**87.30**	**0.866**	**0.954**	0.184	0.310	**0.277**
	FLUX.2-klein-9B	0.770	85.13	0.834	0.888	**0.234**	**0.414**	0.247

Training settings. The cross-model experiments use imitation without repulsion and the same Pi-Flow prompt set as the image-generation experiments. All four cross-model students are evaluated after 2,000 iterations at 4 NFE. Student LoRA rank and alpha are both 256, with dropout 0.05. The student and Pivot learning rates are 10^{-4}, with a 100-iteration warm-up, four teacher queries per segment, and four samples per GPU. Pivot uses four coefficient vectors for the polynomial path basis. FLUX.2 teacher targets use CFG 4, while Lens self-distillation uses CFG 5 with its native norm rescaling.

As shown in Table 14, the 9B FLUX.2 teacher improves the 4B FLUX.2 student on GenEval, DPG-Bench, alignment, text rendering, diversity, and style, with a small decrease in reasoning. The cross-architecture result is more mixed: using FLUX.2-klein-9B to supervise Lens-RL-3.8B improves OneIG diversity and style, but self-distillation remains stronger on the other reported metrics. Cross-model supervision is therefore feasible, but a larger teacher is not uniformly better when its architecture and output distribution differ from those of the student.

Figure 17 compares outputs for seasonal composition, portrait, laboratory, and stylized-character prompts. Changing the teacher can alter composition and style even when the target student architecture and inference budget are fixed.

F.2 Multiple Experts as Teachers

A single teacher may not be strongest on every criterion. We therefore study whether one 4-NFE SD3.5-Medium student can absorb supervision from four reward-specialized LoRA teachers, targeting CLIPScore, GenEval, OCR, and PickScore. All students share the same initialization and use a timestep shift of 3.0. This experiment uses classifier-free-guided teacher targets and an imitation-only objective without repulsion.

We construct four 30k-prompt routes from the corresponding Flow-GRPO (Liu et al., 2025a) training splits with LLM expansion and combine them into a 120k-entry pool. Self-distillation samples



Figure 17. **Changing the teacher changes the student outputs.** From left to right, each pair of columns shows FLUX.2-klein-9B → 4B, FLUX.2-klein-4B → 4B, FLUX.2-klein-9B → Lens-RL-3.8B, and Lens-RL-3.8B self-distillation. Each pair contains two 4-NFE generations for the same prompt.

Table 15. SD3.5-Medium distillation from one or multiple teachers. Specialist rows are 4-NFE students, not teachers; bold and underline mark the best and second-best student scores.

Setting	NFE	GenEval↑	OCR↑	PickScore↑	HPSv2↑	CLIPScore↑	ImageReward↑
Teacher	28	0.6279	0.5813	21.76	0.2802	0.2837	0.8512
Self-distill	4	0.6153	0.3019	21.60	0.2848	0.2741	0.8163
CLIPScore specialist	4	0.7082	0.4202	22.22	<u>0.3055</u>	**0.3011**	**1.223**
GenEval specialist	4	**0.7385**	0.3153	21.53	0.2658	0.2799	0.9196
OCR specialist	4	0.6410	<u>0.5944</u>	21.71	0.2841	0.2803	0.9002
PickScore specialist	4	0.6551	0.3738	**22.81**	**0.3107**	0.2706	1.120
Routed multi-teacher	4	<u>0.7289</u>	**0.5948**	<u>22.56</u>	0.2995	<u>0.2848</u>	<u>1.141</u>

from the combined pool with the base teacher. Each single-teacher specialist uses one fixed reward teacher across all four routes, while routed multi-teacher distillation selects the corresponding reward teacher for each prompt route. The self-distilled and multi-teacher students are evaluated after 3,500 iterations, and the four single-teacher students after 2,500 iterations.

Training settings. For SD3.5-Medium, the student is trained at full rank with learning rate 10^{-5}, and one shared Pivot uses learning rate 10^{-4}. Teacher CFG is 5, with two query points per segment and four samples per GPU. Multi-teacher training samples the four routes with equal probability and uses the corresponding reward teacher for each route-homogeneous batch.

We evaluate each capability on its corresponding prompt distribution: GenEval, OCR, and PickScore use their respective test sets, while CLIPScore, HPSv2, and ImageReward are measured on DrawBench. Thus, the six columns in Table 15 summarize complementary evaluations rather than six measurements on one common prompt set. Individual specialists transfer different strengths: the GenEval specialist gives the highest GenEval score, the PickScore specialist leads on PickScore and HPSv2, and the CLIPScore specialist leads on CLIPScore and ImageReward. Routed multi-teacher distillation gives the best OCR score and the second-best GenEval, PickScore, CLIPScore, and ImageReward scores among the students. It does not dominate every metric, but provides the most balanced result across the reported evaluations.



Figure 18. **Multiple experts supervise one student.** SD3.5-Medium columns show the multi-step base teacher and 4-NFE students trained by self-distillation, four individual reward specialists, or routed multi-teacher supervision. Each row uses the same prompt across columns; routing is used only during training.

Figure 18 illustrates how the supervision source changes text rendering, composition, and visual style for fixed prompts. The routed student remains coherent across the shown prompt types, complementing its balanced quantitative result.

G Additional Ablation Results

This appendix supplements Section 5 with prompt-coverage experiments, the complete Qwen-Image metric breakdowns, and qualitative comparisons for both backbones. We use the arithmetic mean of the five OneIG dimensions as Overall; this is our summary statistic, not an official OneIG aggregate. VBench Overall is $0.8\,\mathrm{Quality} + 0.2\,\mathrm{Semantic}$. Aggregates are computed before rounding, and bold marks the best value within each comparison group. Each setting represents one training run; the four generation seeds shown in qualitative comparisons are not independent training repetitions.

G.1 Prompt Coverage

We first train Qwen-Image students on 30K prompts from each of four categories: Text, Portrait, Objects, and Nature. A Mixed student uses all four categories, with 120K prompts in total. Table 16 reports both the category comparison and the full support-size sweep, including every metric dimension. The text-focused set gives the strongest Overall among these category settings, but no category leads every dimension. Figure 19 compares their responses to the same two test prompts. Training category affects both the requested content and its appearance. Table 16 shows that the largest differences among the 30K settings occur in Text, while Alignment remains comparatively close. Mixed changes both category coverage and support-set size, so its difference from a single-category student cannot be attributed to category mixing alone.

Table 16. Qwen-Image prompt coverage across training categories and support-set sizes.

Category	Prompts	Alignment	Text	Reasoning	Style	Diversity	Overall
Text	1	0.838	0.939	0.278	0.370	**0.265**	0.538
Text	300	0.878	**0.966**	0.305	0.420	0.134	0.541
Text	3K	**0.881**	0.964	**0.307**	0.418	0.129	0.540
Text	30K	0.876	0.961	0.299	**0.421**	0.150	**0.541**
Portrait	1	0.847	0.610	0.239	0.411	**0.310**	0.484
Portrait	300	0.886	**0.926**	0.290	**0.442**	0.155	**0.540**
Portrait	3K	0.883	0.881	0.292	0.428	0.142	0.525
Portrait	30K	**0.889**	0.903	**0.295**	0.439	0.158	0.537
Objects	1	0.854	0.636	0.242	0.431	**0.303**	0.493
Objects	300	0.887	**0.912**	**0.301**	**0.437**	0.134	**0.534**
Objects	3K	**0.889**	0.903	0.295	0.434	0.145	0.533
Objects	30K	0.889	0.907	0.297	0.429	0.137	0.532
Nature	1	0.849	0.407	0.237	0.344	**0.253**	0.418
Nature	300	0.883	0.804	0.291	0.426	0.160	0.513
Nature	3K	**0.886**	**0.852**	**0.298**	0.424	0.137	**0.519**
Nature	30K	0.884	0.830	0.283	**0.429**	0.159	0.517
Mixed	120K	0.885	0.938	0.300	0.432	0.138	0.538

Table 17. Pivot correction and initialization on Qwen-Image. Active branches use $K = 4$, $M = 3$.

Initialization	Alignment	Text	Reasoning	Style	Diversity	Overall
Zero residual	0.886	0.937	0.300	0.424	0.127	0.535
Pretrained	0.885	0.938	0.300	0.432	**0.138**	0.538
Random	**0.887**	**0.944**	**0.303**	**0.438**	0.135	**0.541**

Table 18. Feature-tap position K and Pivot depth M on Qwen-Image.

K	M	Alignment	Text	Reasoning	Style	Diversity	Overall
1	3	**0.889**	0.943	**0.305**	0.433	0.130	0.540
4	3	0.887	**0.944**	0.303	**0.438**	**0.135**	**0.541**
1	7	0.886	0.945	0.302	0.430	0.108	0.534
8	7	**0.891**	**0.967**	**0.310**	**0.434**	**0.132**	**0.547**

Table 19. Residual basis family and size P on Qwen-Image with other Pivot settings fixed.

Basis	P	Alignment	Text	Reasoning	Style	Diversity	Overall
Polynomial	4	0.885	0.938	0.300	**0.432**	0.138	0.538
Spectral	4	0.887	0.943	**0.307**	0.427	0.129	0.538
Collocation	4	0.881	0.928	0.291	0.422	0.143	0.533
Piecewise	4	0.884	0.944	0.300	0.429	**0.145**	0.540
Piecewise	32	**0.889**	**0.962**	0.306	0.426	0.132	**0.543**

Table 20. Reference guidance w and repulsion weight λ on Qwen-Image at teacher guidance 5.

w	λ	Alignment	Text	Reasoning	Style	Diversity	Overall
–	0	0.883	0.943	0.295	**0.434**	**0.142**	0.539
0	0.1	**0.888**	**0.960**	0.298	0.429	0.123	0.539
0	0.2	0.886	0.958	**0.304**	0.421	0.139	**0.542**
1	0.1	0.883	0.899	0.294	0.425	0.138	0.528
1	0.2	0.885	0.938	0.300	0.432	0.138	0.538

Mixed combines four 30K-prompt categories. Bold marks the best score within each comparison group, with feature-tap comparisons grouped by fixed M.

Within each category, we reduce the support set to 3K, 300, or one prompt. The transition from one to 300 prompts substantially improves Text for the Portrait, Objects, and Nature students. Further increases do not improve all dimensions monotonically. The higher Diversity scores of several single-prompt students may include semantic drift, rather than useful variation that preserves the



Figure 19. **Training prompt coverage shapes generation.** Qwen-Image columns show Text-30K, Portrait-30K, Objects-30K, Nature-30K, and Mixed-120K, with four samples per cell. The crystal-dragon prompt (top) is a new test case outside OneIG scoring; the Vertex Consulting brochure (bottom) is a OneIG case. A readable brochure does not imply that every requested word is reproduced correctly.



Figure 20. **Unseen prompts after single-prompt distillation.** Four Qwen-Image students trained on one Text, Portrait, Objects, or Nature prompt are evaluated on unseen paper/ribbon/wax-seal and sofa-portrait prompts. Each cell contains four samples. The final column is the native Base 4-step reference with its own sampling settings.

requested content. OneIG Text is also a composite rather than a word-accuracy measure: in the displayed Text-30K brochure case, WAC is 1.0 while CR is only 0.25.

Figure 20 compares four students trained on one prompt each. Their responses to two unseen prompts differ in content and appearance. The four training prompts also differ in length and number of constraints, so these examples do not isolate a mechanism behind the differences. The native Base 4-step model uses its own CFG and time schedule and serves as a visual reference.

We also inspect three Wan students trained for 200 iterations on a running-boy, running-dalmatian, or post-rain desert-bloom prompt (Figure 21). Each is evaluated on its training condition and four unseen conditions. In the desert-bloom row, dot-like visual patterns remain visible under other prompts, suggesting that appearance associated with the training condition can carry into unseen conditions. These runs have no corresponding VBench scores.

Prompt category, support-set size, and the content of an individual training prompt all affect the observed outputs. The results do not identify a universally sufficient number of distillation prompts.



Figure 21. **Cross-prompt behavior after single-prompt video distillation.** Wan2.1-1.3B rows use the running-boy, running-dalmatian, and desert-bloom training prompts. The boxed GT column is a Wan14-generated training reference, not real-world ground truth. Seen Prompt shares its condition, but not a paired initial noise trajectory. The other columns use unseen prompts. Each cell takes the top-left sample of the video grid, at frame 41/81 for Park Walk and 77/81 otherwise, consistently across rows.

G.2 Pivot Design

Experimental scope. The design sweeps vary initialization, feature-tap position K, Pivot depth M, or residual basis size P within each model's ablation recipe. The feature-location and depth comparisons use random initialization; the initialization comparison fixes $K = 4$ and $M = 3$. Qwen design comparisons use iteration-1,000 checkpoints evaluated on OneIG, while the Wan design study uses iteration-300 checkpoints under the PDD-public protocol. For Wan, the common recipe uses rank/alpha-256 LoRA, four-step Euler sampling with timestep shift 6, two query points per segment, and student/Pivot learning rates of 10^{-4} with 100 warm-up iterations. The default reference has $w = 0$ and $\lambda = 0.1$, with teacher guidance 5; the basis sweep fixes pretrained initialization at $K = 4$, $M = 3$. Wan runs retain the same 250K-entry training dataset and deterministic data order, with trainer and student LoRA seeds fixed at zero. These are separate from the main-comparison checkpoints in Appendix H: the reported Qwen model uses Piecewise with $P = 4$, reference guidance $w = 1$, and $\lambda = 0.2$, while the main Wan models are evaluated at iteration 200. The best setting within an ablation sweep should therefore not be read as the configuration used in every main comparison.

Correction and initialization. Table 17 adds the full Qwen dimensions to the initialization comparison in Table 5. The complete Wan initialization scores already appear in Table 6. Qwen's no-correction control executes Pivot with its residual disabled, whereas Wan skips the Pivot forward pass; both retain the student's velocity head.

Feature location and capacity. Table 18 reports Qwen's full metrics for the fixed-M feature comparisons and also contains the $K = 1$, $M \in \{3, 7\}$ depth comparison. Wan's corresponding scores are in Table 6. Figures 22 and 23 show the selected qualitative comparisons at fixed capacity.

Figure 24 complements the $K = 1$ rows of Table 18 with five paired prompt comparisons. This decoder-depth comparison changes M, not the number of residual basis elements P.

Residual basis. Table 19 gives the Qwen metric breakdown for the basis comparison in Section 5; the complete Wan scores are in Table 6. Piecewise with $P = 32$ improves Qwen Alignment and Text relative to $P = 4$, but decreases Diversity. The basis comparison concerns the representation of the residual, distinct from the decoder-depth comparison above.

G.3 Teacher Repulsion

We fix teacher guidance at 5 and use a pretrained Pivot with $K = 4$ and $M = 3$ and the Polynomial basis with $P = 4$. We vary only the reference CFG scale w and the repulsion weight λ. Here $w = 0$ gives an unconditional reference and $w = 1$ gives a conditional reference without amplified guidance. At $\lambda = 0$, repulsion is disabled and w has no effect. All settings retain an active Pivot correction.

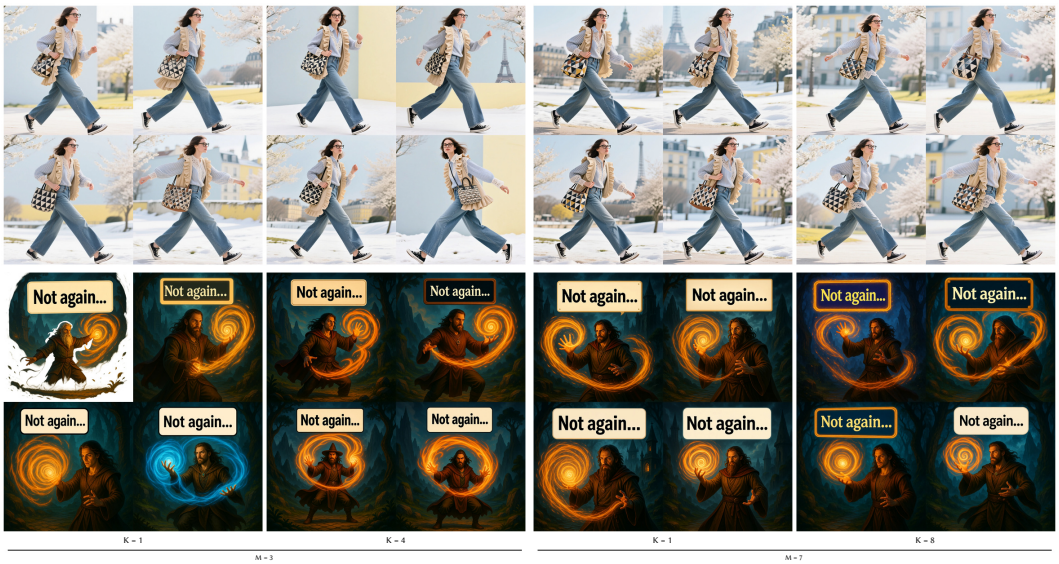


Figure 22. **Feature-tap location at fixed Pivot depth.** Qwen-Image outputs for spring-fashion and wizard prompts compare $K = 1, 4$ at $M = 3$ (left) and $K = 1, 8$ at $M = 7$ (right). Each cell shows four samples; Pivot depth is fixed within each group.



Figure 23. **Feature-tap location in video distillation.** Wan2.1-1.3B columns compare $K \in \{1, 4, 8\}$ at fixed Pivot depth $M = 3$. Rows show golf, night-market lanterns, and butterflies, at frames 77/81, 77/81, and 65/81, respectively. Each cell contains four samples. Lanterns and butterflies use standalone iteration-300 inference with matching prompt and noise indices. Golf uses the training callback with consistent base seed and sample order, without verified tile-level noise equality. The still frames illustrate appearance, not full-video motion quality.



Figure 24. **Pivot depth at a fixed feature tap.** Qwen-Image uses $K = 1$, with $M = 3$ in the top group and $M = 7$ in the bottom group. Columns show mushroom fashion, Tokyo neon, a mountain factory, a museum entrance, and a home advertisement. Each column uses the same prompt and four corresponding generation seeds.

The complete Wan sweep is reported in Table 8, including the no-repulsion control and $w \in \{0, 1\}$ at $\lambda \in \{0.1, 0.2, 0.4\}$. Table 20 adds Qwen's Alignment, Reasoning, and Style dimensions to the main-table comparison.

For Qwen, Table 20 compares both references at the shared weights $\lambda \in \{0.1, 0.2\}$. With $w = 0$, increasing the weight improves Reasoning and Diversity but reduces Style. At $\lambda = 0.1$, the $w = 0$ reference also gives higher Text than $w = 1$. Reference choice and strength therefore have distinct effects and should not be collapsed into a single ranking of repulsion settings.

Figures 25 and 26 retain every setting for each selected case, including the no-repulsion reference. They supplement the aggregate metrics with changes in appearance and composition; still frames alone do not establish motion quality.

H Few-step Implementation and Evaluation Details

Our distillation runs use eight NVIDIA A100 GPUs. The settings below describe the checkpoints used in the reported experiments.

H.1 Image Generation

We use the text prompt set provided by Pi-Flow (Chen et al., 2026) and group training samples into resolution buckets for variable-resolution distillation. Training uses only text prompts and target image dimensions, with student trajectories generated on-policy from random noise.

For FLUX.1-dev and Qwen-Image, the student is initialized from its corresponding teacher and trained with rank-256 LoRA adapters. For both models, Pivot copies the final three blocks and predicts four residual coefficient vectors, with two teacher query points per segment. FLUX.1-dev uses the polynomial path basis, while Qwen-Image uses the Piecewise basis. The target and weaker-reference guidance scales are 3.5 and 1.0 for FLUX.1-dev's embedded guidance, and 5.0 and 1.0 for Qwen-Image's classifier-free guidance. The corresponding repulsion weights are 0.4 and 0.2.

The global batch sizes are 32 for FLUX.1-dev and 16 for Qwen-Image, without gradient accumulation. Both the student and Pivot use a learning rate of 10^{-4}, and EMA is disabled. For FLUX.1-dev, we use AdamW with zero weight decay and a 100-iteration linear warm-up followed by a constant learning rate. We select checkpoints based on their combined performance on GenEval, DPG-Bench, and OneIG-Bench-EN. The reported FLUX.1-dev and Qwen-Image checkpoints are taken after 1,250 and 1,000 training iterations, respectively, and evaluated at 4 NFE.



Figure 25. **Reference choice and repulsion strength.** The five-setting Qwen-Image comparison varies reference guidance w and repulsion weight λ. Cases, in reading order, are girl and dragon, laboratory scientist, mechanical age, and forest signs. Within each case, columns vary λ and rows vary w, with four corresponding generation seeds per cell. Because w is inactive at $\lambda = 0$, the black frame encloses two displays of the same baseline. Labels are shared across the two case rows or columns. OneIG Text is case-sensitive, so readable text with different capitalization may still incur a penalty.

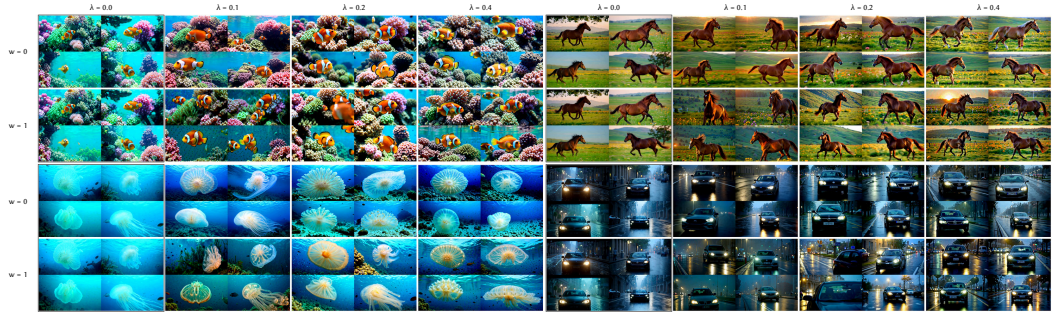


Figure 26. **Repulsion settings in video distillation.** The seven-setting Wan2.1-1.3B comparison uses iteration-300 checkpoints. Cases, in reading order, are clownfish, running horses, jellyfish, and a car on a rainy night. Each cell uses VBench samples 0–3 at frame 77/81, with prompts and sample indices aligned across settings. The black frame encloses the repeated $\lambda = 0$ baseline; shared labels follow Figure 25. Initial noise was not verified pixel by pixel. Static frames do not measure full-video motion quality.

H.2 Image Editing

We collect source images paired with short English editing instructions covering object editing, appearance changes, style transfer, and text editing. Student trajectories are generated on-policy from random noise, and target edited images are not used for supervision. We follow the official conditioning pipeline, encoding source images with the VAE posterior mode and assigning their conditioning tokens a timestep of zero.

The Qwen-Image-Edit-2511 student is initialized from its teacher and trained with rank-256 LoRA adapters. Pivot copies the final three blocks and predicts eight residual coefficient vectors for the polynomial path basis, with two teacher query points per segment. The target and weaker-reference guidance scales are 5.0 and 1.0, respectively, with a repulsion weight of 0.4. We retain the model's native CFG norm rescaling and use dynamic timestep shifting with a configured shift factor of 2.0.

The global batch size is 16 without gradient accumulation, and both the student and Pivot use a learning rate of 10^{-4}. We report the checkpoint after 250 training iterations on both editing benchmarks.

Our student and the released Lightning checkpoint are evaluated at 4 NFE under matched settings. ImgEdit-Bench uses GPT-4o to score editing operations, while GEdit-Bench-EN uses GPT-4.1 to score semantic consistency, perceptual quality, and overall quality.

H.3 3D Shape and Texture

For Hunyuan3D 2.1, training uses one rendered view per object from a TexVerseFull-derived collection. We preserve the image alpha channel and apply the official recentering and white-background compositing pipeline. The image provides the conditioning for an on-policy rollout in the shape latent space, without using ground-truth shape latents for supervision.

The shape student is initialized from its teacher and trained with rank-128 LoRA adapters and a trainable output normalization layer. The latent representation consists of 4,096 vectors with 64 channels each. A full-rank Pivot copies the final three blocks and output layer, predicting four residual coefficient vectors with four teacher query points per segment. The teacher guidance scale is 5.0, and repulsion uses the conditional teacher prediction as its reference with weight 0.2. We also use an unconditional-imitation ratio of 0.2. The global batch size is 32 without gradient accumulation. The illustrated checkpoint is taken after 1,500 training iterations and evaluated at 4 NFE for shape generation.

For TRELLIS.2, we independently distill the high-resolution shape and texture models in the 1024-resolution cascade, retaining the pretrained sparse-structure and coarse-shape stages. Training uses conditioning images and sparse voxel coordinates from a TexVerseFull-derived collection. The coordinates provide spatial support for on-policy rollouts initialized from random noise, without using ground-truth sparse features to construct off-policy training states. The texture model is additionally conditioned on the shape latent.

Both students are initialized from their corresponding pretrained models and trained with rank-128 LoRA adapters, with the output normalization layers also trainable. Each full-rank Pivot copies the final three DiT blocks and predicts four residual coefficient vectors for the polynomial path basis, with four teacher query points per segment. We use the imitation objective without repulsion. The shape teacher uses guidance scale 7.5 over the interval $[0.6, 1.0]$ with guidance rescaling of 0.5, while the texture teacher uses guidance scale 1.0 without rescaling.

Both TRELLIS.2 stages use AdamW with zero weight decay and learning rates of 10^{-5} for the student and 10^{-4} for Pivot. A 100-iteration linear warm-up is followed by constant learning rates, and EMA is disabled. The global batch size is 32 with two gradient-accumulation rounds. We use a timestep shift of 3.0 and report the checkpoints after 500 training iterations for each stage. The 4-NFE inference budget applies separately to the distilled shape and texture stages.

H.4 Video Generation

We use a collection of 248K preprocessed dataset provided by rCM (Zheng et al., 2026) for Wan2.1 distillation. Training uses on-policy student rollouts, with two teacher query points per segment sampled in sigma space.

Both students are initialized from their corresponding teachers and trained with rank-256 LoRA adapters. For the 1.3B model, Pivot copies the final three blocks and predicts four residual coefficient vectors for the polynomial path basis. The target and weaker-reference guidance scales are 5.0 and 0.0, respectively, with a repulsion weight of 0.1. The selected 14B checkpoint uses zero active Pivot residual, with target and weaker-reference guidance scales of 6.0 and 1.0 and a repulsion weight of 0.2. When computing the unconditional velocity, we skip the layer at index 9 for Wan2.1-T2V-1.3B and index 11 for Wan2.1-T2V-14B, following the layer indices in the implementation. Layer skipping has been used by the Wan community to improve generation quality and has also been explored in prior work (Shaul et al., 2026; Hyung et al., 2024). The timestep shift factors are 10 for the 1.3B model and 5 for the 14B model.

Both models use a global batch size of 16 without gradient accumulation and a student learning rate of 10^{-4}. The Pivot learning rates are 10^{-4} for the 1.3B model and 3×10^{-5} for the 14B model, and EMA is disabled. For each backbone, we select the checkpoint based on the overall VBench score and report the model after 200 training iterations.

VBench evaluation uses 944 Self-Forcing prompts, generating five videos per prompt at 832×480 resolution, with 81 frames at 16 FPS. Generation uses expanded prompts, while scoring uses the corresponding short prompts. All students are evaluated at 4 NFE. The released baselines are evaluated under the same settings, while PDD results are taken from its paper (Shaul et al., 2026). We report the standard Quality and Semantic aggregates separately, together with the individual VBench dimensions.

H.5 Detailed VBench Results

Tables 21 and 22 report all 16 VBench dimensions for the models compared in Table 3. The Quality and Semantic aggregates use VBench's standard per-dimension normalization and weights.

Dynamic Degree uses RAFT optical flow to identify videos with sufficiently large motion across multiple frame pairs, and reports the fraction classified as dynamic. Motion Smoothness measures agreement between held-out frames and frames reconstructed by a video interpolation model, with lower reconstruction error yielding a higher score. We report these dimensions together to distinguish motion activity from temporal smoothness. Color measures whether detected objects exhibit the colors specified by the prompts, rather than color saturation or visual naturalness. These per-video dimensions do not directly measure diversity across repeated generations of the same prompt.

H.6 Joint Audio–Video Generation

We initialize the students from MiniMax H3, a 50-block joint audio–video Transformer, and keep the teacher frozen. The base teacher is already guidance-distilled, so no additional classifier-free guidance branch is used. Text and multimodal conditions are encoded using layer-50 hidden states from Qwen3-VL-32B and cached in BF16. For text-conditioned generation, we collect 100,000 structured prompts describing visual content, soundscapes, and non-diegetic music. Training uses text conditions only, without first- or last-frame inputs or real target audio–video pairs. Student trajectories are generated on-policy from random noise.

Both the student and Pivot use LoRA rank and alpha 128, with AdamW at a learning rate of 10^{-4} and BF16 training. Video and audio imitation losses receive equal weights, with one teacher query point per segment. We use the Piecewise basis with 32 residual coefficient vectors for the 4-NFE student and four for the 8-NFE student. Both configurations use a repulsion weight of 0.2. For both inference budgets, the repulsion reference is computed at the query point by skipping the 25th teacher layer, corresponding to implementation index 24. Layer skipping is applied only to the training-time reference prediction, not to the deployed student. The reported checkpoints are taken after 1,250 and 850 training iterations, respectively. Video and audio timestep shifts are 12 and 3 for our students,

Table 21. VBench quality dimensions for Wan2.1-T2V at 4 NFE. Columns report Subject Consistency (SC), Background Consistency (BC), Temporal Flickering (TF), Motion Smoothness (MS), Dynamic Degree (DD), Aesthetic Quality (AQ), and Imaging Quality (IQ). All metrics are higher-is-better. Boldface and underlining indicate the best and second-best results within each backbone. † marks PDD results quoted from the original paper (Shaul et al., 2026) (Euler variant at 1.3B).

Method	SC	BC	TF	MS	DD	AQ	IQ
Wan2.1-T2V-1.3B							
DMD2	<u>97.76</u>	**97.82**	**99.42**	<u>98.81</u>	45.00	**68.80**	**69.46**
rCM	94.72	95.79	98.77	97.64	**81.94**	64.68	66.73
AnyFlow	**97.79**	<u>96.96</u>	99.26	98.29	68.33	<u>67.01</u>	67.63
PDD†	97.69	95.90	<u>99.33</u>	**99.02**	72.50	66.26	<u>69.03</u>
Ours	96.29	95.44	98.83	98.66	<u>79.44</u>	66.19	68.14
Wan2.1-T2V-14B							
DMD2	<u>96.92</u>	**97.83**	**99.67**	**98.76**	49.72	<u>67.07</u>	67.04
rCM	95.65	96.45	98.57	97.81	<u>76.39</u>	66.76	**69.17**
AnyFlow	**97.22**	<u>97.40</u>	99.00	98.10	73.61	**67.12**	68.78
PDD†	95.41	96.42	<u>99.39</u>	<u>98.52</u>	74.72	66.78	<u>68.88</u>
Ours	96.60	95.73	97.86	<u>98.52</u>	**83.89**	66.47	68.50

Table 22. VBench semantic dimensions for Wan2.1-T2V at 4 NFE. Columns report Object Class (OC), Multiple Objects (MO), Human Action (HA), Color (C), Spatial Relationship (SR), Scene (S), Appearance Style (AS), Temporal Style (TS), and Overall Consistency (OC′). All metrics are higher-is-better. Boldface and underlining indicate the best and second-best results within each backbone. † marks PDD results quoted from the original paper (Shaul et al., 2026) (Euler variant at 1.3B).

Method	OC	MO	HA	C	SR	S	AS	TS	OC′
Wan2.1-T2V-1.3B									
DMD2	<u>95.85</u>	<u>87.80</u>	95.60	84.63	73.76	53.21	**21.59**	23.43	26.28
rCM	93.76	85.64	<u>96.00</u>	85.39	74.95	<u>56.87</u>	20.50	**24.38**	<u>26.37</u>
AnyFlow	95.74	**90.47**	**97.00**	<u>87.67</u>	**80.33**	**57.50**	<u>20.65</u>	<u>24.37</u>	**26.53**
PDD†	**96.22**	85.79	93.34	85.41	<u>76.95</u>	51.47	19.33	23.77	25.73
Ours	94.21	83.84	<u>96.00</u>	**89.19**	74.54	55.11	19.91	23.76	25.82
Wan2.1-T2V-14B									
DMD2	97.17	88.81	97.00	**87.81**	73.55	52.54	**21.88**	23.56	<u>26.50</u>
rCM	**97.59**	90.48	**98.80**	82.91	<u>83.05</u>	53.27	<u>21.00</u>	<u>24.48</u>	26.39
AnyFlow	<u>97.56</u>	**92.50**	<u>98.20</u>	85.79	**83.84**	**54.72**	20.90	**24.68**	**26.83**
PDD†	96.55	87.90	96.40	<u>86.59</u>	80.02	<u>54.30</u>	20.86	24.39	26.49
Ours	97.37	<u>90.70</u>	96.20	86.18	80.92	53.21	19.93	24.10	26.42

compared with 6 and 3 for the DMD baselines. Each method retains its own sampling recipe, so the comparison does not isolate the training objective under an identical sampler.

For reference-conditioned generation, we construct 20,000 instructions paired with 12,000 unique reference audio–video clips from UltraVideo, covering ten reference-role task categories. The reference clips provide conditioning rather than ground-truth target outputs. Target durations of 5, 8, and 10 seconds account for 10,000, 8,000, and 2,000 instructions, respectively. The student uses the same LoRA and optimizer settings, with 32 Piecewise residual coefficients and a repulsion weight of 0.2. Repulsion is evaluated at the query point by skipping the 25th teacher layer, corresponding to implementation index 24. This layer is skipped only in the training-time repulsion branch, not in the deployed student. Training uses eight GPUs with context-parallel degree four, one sample per data-parallel replica, and no gradient accumulation, giving a global batch size of two. The examples in Figure 16 use the checkpoint after 1,000 iterations at 4 NFE. Although the instructions span multiple task categories, the training references are predominantly audio–video clips and do not cover the model's full mixed-reference input space.

Table 23. Paired H3 score differences (ours minus DMD) with 95% bootstrap confidence intervals. Each inference budget contains 100 matched pairs. The two budgets use different evaluation sets. Audio clarity and audio quality denote PAM and the Audiobox Production Quality (PQ) score, respectively.

Metric	8 NFE	4 NFE
Structural coherence	$+0.020$ $[-0.050, +0.090]$	-0.010 $[-0.105, +0.080]$
Temporal coherence	$+0.065$ $[-0.005, +0.140]$	$+0.025$ $[-0.085, +0.140]$
Visual naturalness	$+0.045$ $[-0.025, +0.120]$	$+0.040$ $[-0.040, +0.120]$
Audio clarity	$+0.234$ $[+0.192, +0.277]$	$+0.269$ $[+0.230, +0.309]$
Audio quality	$+0.152$ $[+0.069, +0.230]$	$+0.430$ $[+0.285, +0.578]$

H.7 H3 Evaluation and Additional Results

The 8-NFE comparison uses all 100 prompts in the UltraVideo evaluation set and the released DMD v1.1 checkpoint. The 4-NFE comparison uses DMD v1.1 and a separate five-second set with 20 prompts in each of five categories: short dialogue, continuous takes, two-shot sequences, camera reveals, and large motion. Prompts, seeds, conditioning inputs, output resolution, and duration are matched within each pair. The 8-NFE set includes prompts previously used for evaluator calibration and development, so it is not an entirely held-out test set.

Visual evaluation uses GPT-5.6 Sol with 24 temporally distributed paired frames and two independent evaluations with reversed candidate order. The judge assigns 1–5 scores for structural coherence, temporal coherence, and visual naturalness, which are averaged across candidate orders and prompts. The rubric permits an equivalent verdict unless persistent, material differences are present, and does not reward sharpness, saturation, or contrast alone. It does not score prompt completeness. Combining strict and weak preferences gives 13/80/7 judgments for ours/equivalent/DMD at 8 NFE and 22/58/20 at 4 NFE. Two-sided sign tests give $p = 0.263$ and $p = 0.878$, respectively. Here, equivalent is a judge category, not the outcome of a statistical equivalence test.

Audio clarity is measured using PAM (Deshmukh et al., 2024), which contrasts descriptions of clean audio with descriptions of noisy, artifact-laden audio. We aggregate its scores over fixed seven-second chunks. Audio quality is measured using the Production Quality (PQ) score from Audiobox Aesthetics (Tjandra et al., 2025). These labels identify the two existing metrics in our figures and tables; their scoring procedures are unchanged. Table 23 reports paired mean differences and bootstrap confidence intervals for all five metrics. Both audio intervals exclude zero at each budget, whereas all three visual intervals include zero. The intervals describe variation across the evaluated prompt pairs with fixed evaluators, not uncertainty from model selection or evaluator development. These automatic evaluations do not establish human preference or audio–visual synchronization quality.

Figure 27 provides selected 4-NFE examples with continuous spectrograms of their generated soundtracks. The illustrated student uses the same skip-layer repulsion configuration and 1,250-iteration checkpoint as the 4-NFE model evaluated in Table 4. Spectrograms use native 32 kHz stereo audio, a 1,024-sample Hann window, and a 128-sample hop. We average channel powers after the short-time Fourier transform and apply a shared $[-90, 0]$ dB scale over 0–16 kHz, without per-clip normalization or denoising. The spectra cover the full clip and illustrate the soundtrack structure, rather than serving as a perceptual-quality metric.

All H3 qualitative panels use selected examples rather than a randomly sampled evaluation subset. The reference-conditioned cases were selected after prompt and seed exploration. They illustrate retained conditioning behavior, without establishing an aggregate success rate or exact motion and geometry correspondence. The displayed examples retain imperfections in expression timing, instrument-playing gestures, and fine-grained motion.

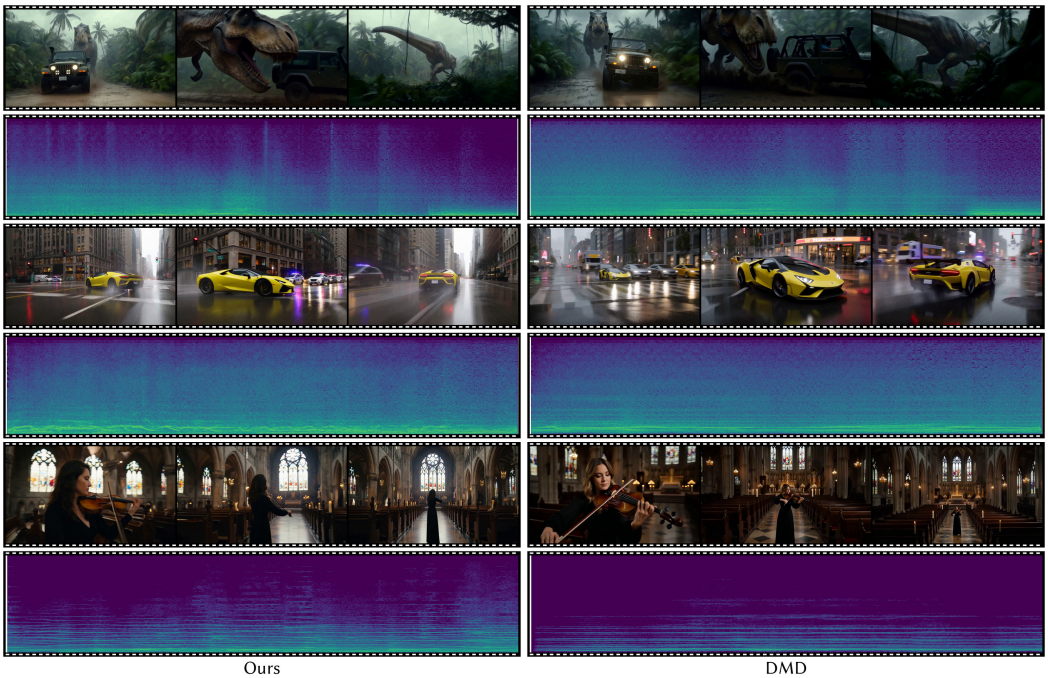


Figure 27. **Video frames and generated soundtracks.** Selected H3 examples at 4 NFE compare ours (left) with DMD v1.1 (right). From top to bottom, the cases depict a tyrannosaur pursuit, a supercar drift, and a violin performance. Three chronological video frames alternate with continuous full-clip audio spectrograms on a shared frequency and power scale. The illustrated student is the 4-NFE model evaluated in Table 4.