StepX-OPD: Rethinking Multi-Teacher On-Policy Distillation in Diffusion Models

Anonymous authors
Paper under double-blind review

Abstract

Text-to-image diffusion models must combine broad capabilities, such as instruction following, text rendering, perceptual quality, and human preference, with the few-step sampling that deployment requires. On-policy distillation supervises a student on states it generates itself, consolidating capabilities from several specialized teachers into one model, while step distillation compresses a sampling trajectory into a few steps; we study combining the two. However, existing pipelines instead chain them: a full-step multi-teacher model is trained first and compressed only afterward, so compression waits on that stage, and the capabilities it consolidates are not guaranteed to survive compression. To address this, we introduce StepX-OPD, a multi-teacher on-policy distillation method that jointly consolidates specialist capabilities and compresses the sampling trajectory: after a brief self-distillation warmup, each mini-batch is drawn from one training source and routed to the frozen reward-specialized teacher for that source, which drives distribution matching on states supplied by the student's own few-step rollout. Furthermore, on low-noise batches the routed teacher also regresses the student's velocity field directly, balancing capability transfer against the fidelity of the few-step trajectory itself. Baseline evaluations span SD3.5-Medium and FLUX.1-dev, scoring GenEval, OCR, DeQA, PickScore, and their normalized average. The quantitative result reported here for StepX-OPD is on SD3.5-Medium, where it attains the highest Average among the four-step methods compared and exceeds the Average of the forty-step Flow-OPD model. Jointly optimizing capability integration and step compression on one student trajectory avoids using a consolidated full-step model as the teacher for a later compression stage.

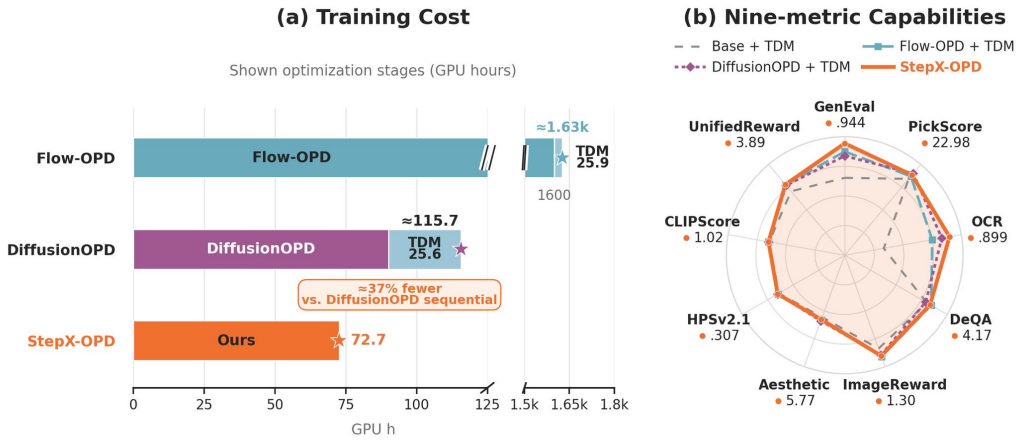


Figure 1: Training cost for the displayed optimization stages (left) and four-step performance across nine benchmarks after fixed axis-wise normalization (right). GPU-hours were measured on 24 NVIDIA H800 GPUs.

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

Flow matching and diffusion models are widely used for text-to-image generation (Lipman et al., 2023; Esser et al., 2024). A model prepared for deployment must meet two needs at once: it must be capable across composition, text rendering, image quality, and human preference, and it must be fast, measured by how few sampling steps its inference requires. Models meet these needs through two separate lines of post-training, one that integrates capabilities into a single model and one that compresses its sampling trajectory into a few steps. These two lines are developed independently, each advancing its own need while leaving the other untouched. Yet both needs are placed on the same deployed model, so capability integration and step distillation constrain that model together.

Existing methods address these two deployment needs separately. Capability integration builds on on-policy distillation, which supervises the student on states the student itself generates (Agarwal et al., 2024). In continuous image generation, DiffusionOPD (Li et al., 2026), Flow-OPD (Fang et al., 2026), and DanceOPD (Zhou et al., 2026) use analytic transition KL or local regression to integrate capabilities, but retain full-step sampling. Step distillation, by contrast, starts from a teacher whose capabilities are already fixed. DMD (Yin et al., 2024b) and DMD2(Yin et al., 2024a) use a frozen teacher score and an auxiliary fake-score estimator, while Qwen-Image-Flash (Wu et al., 2026) combines a base teacher with a post-trained teacher. TDM, CDM, and SenseFlow move supervision toward student-generated trajectories or nearby states, which improves the few-step trajectory. These compression methods still treat teacher capabilities as inputs. A multi-capability few-step model therefore continues to rely on a serial pipeline: multi-teacher capability integration first produces a full-step multi-capability model, and step distillation then treats that model as a fixed teacher for few-step compression. The serial pipeline lengthens training, because compression starts only once the full-step capability model finishes. Fig. 1 reports the training cost each optimization stage carries. Beyond this training cost, it leaves open the question: *do the capabilities integrated into the full-step model still hold once its sampling trajectory is compressed to a few steps?*

We test this directly by comparing each full-step capability model with its few-step compressed counterpart. Tab. 1 reports lower aggregate scores after compression for every evaluated full-step capability model, with larger declines on individual metrics. Fig. 4 shows full-step OPD models, before any compression, rendering prompt words into images more often than the base model on prompts that neither request nor forbid text. Transition matching in continuous-image OPD and marginal matching in DMD (Yin et al., 2024b)/TDM (Luo et al., 2025) both admit reverse-KL interpretations over different distributions and states. Theorem A.2 analyzes this relationship through a reverse-KL decomposition. Motivated by this analysis, we add low-noise velocity MSE (vMSE) to the distribution-matching loss.

To optimize the two stages together, we propose `StepX-OPD`. After a short self-distillation warmup, each mini-batch is drawn from a single training source and routed to the frozen reward specialist for that source. The student then provides continuous states from its current few-step rollout. The selected specialist drives distribution matching at these states, and on batches whose sampled states have low mean noise it also supervises the student's velocity directly. Together, these objectives integrate capabilities while compressing the student, without an intermediate full-step OPD model. This analysis motivates the auxiliary low-noise velocity supervision.

We evaluate `StepX-OPD` on SD3.5-Medium (Esser et al., 2024) and FLUX.1-dev (Labs et al., 2025) across GenEval (Ghosh et al., 2023), OCR accuracy (Chen et al., 2023), DeQA (You et al., 2025), and PickScore (Kirstain et al., 2023). We compare it with sequential Flow-OPD + TDM and DiffusionOPD + TDM baselines. Across this comparison, `StepX-OPD` achieves the highest Average among the evaluated few-step SD3.5 systems, 0.8901. This result demonstrates strong aggregate performance in the final few-step setting.

Our contributions are summarized as follows:

- We introduce `StepX-OPD`, which uses the same selected teacher to jointly train routed OPD and few-step compression on the student's current rollout.
- We measure unintended prompt-word rendering on prompts that neither request nor forbid text, find it rises with capability integration, `StepX-OPD` included, and give a variant that reduces it.
- We evaluate the serial pipeline and `StepX-OPD` under a common protocol across the text-to-image benchmarks. Code will be released with the formal version.

2 RELATED WORK

Step distillation. Within distribution-matching methods, DMD uses the difference between real and fake score fields as a reverse-KL gradient. It also adds regression on teacher-generated noise–image pairs, so part of the update still depends on a multi-step teacher sampler (Yin et al., 2024b). DMD2 removes this paired regression, uses two-time-scale updates to stabilize fake-score learning, adds adversarial training, and uses fixed-schedule backward simulation for multi-step students (Yin et al., 2024a). Decoupled DMD separates CFG augmentation from distribution matching and gives them different re-noising schedules, while retaining DMD2's fixed simulation and fake-score training (Liu et al., 2026a). A data-dependent alternative, Moment Matching, matches conditional clean-data means. SiD derives data-free score identities (Salimans et al., 2024; Zhou et al., 2024). TDM instead matches intermediate trajectory marginals over step-dependent intervals, without instance-level trajectory matching (Luo et al., 2025). SenseFlow adapts DMD to large flow backbones through implicit alignment, intra-segment guidance, real-data supervision, and a VFM discriminator, while keeping coarse fixed anchors (Ge et al., 2025). CDM replaces fixed anchors with dynamically sampled on- and off-trajectory states. This broadens supervision but adds training-time evaluations, while inference uses the same number of steps (Liu et al., 2026b). Qwen-Image-Flash blends base and reinforcement-trained teacher scores with a fixed step schedule. The selected blend is more stable than its specialist-only setting (Wu et al., 2026). These methods still provide fixed capabilities to the distillation objective; Qwen's blend is also fixed in advance rather than chosen from the states the student visits, motivating the on-policy routing methods discussed next.

On-policy distillation in diffusion models. On-policy distillation lets a teacher supervise the states the student visits during generation, rather than states drawn from an offline trajectory or dataset. DiffusionOPD (Li et al., 2026) lifts this idea from the discrete token transitions used in language models to the continuous-state denoising process, modeling each step as a Gaussian transition and deriving a closed-form per-step KL objective in place of a PPO-style surrogate. Flow-OPD (Fang et al., 2026) first trains a domain-specialized teacher per reward with single-reward GRPO, then merges the teachers into a cold-started student via on-policy sampling and task-routing labeling, adding a Manifold Anchor Regularization term in which a task-agnostic teacher anchors generation to a high-quality manifold. DanceOPD (Zhou et al., 2026) treats each frozen capability as a velocity field over the denoising state space, routing each sample to one field, querying it once on a low-noise state from the student's rollout, and training with a plain velocity mean-squared-error objective. A second arrangement applies on-policy supervision after step distillation rather than before it. D-OPSD (Jiang et al., 2026) tunes an already step-distilled few-step model by having the same network act as teacher and student under different conditioning, the student seeing only the text prompt while the teacher also sees the target image, so it learns new concepts or styles on its rollout without degrading its distilled few-step behavior. Both arrangements treat capability integration and step-count reduction as separate stages; `StepX-OPD` places multi-teacher on-policy supervision inside the few-step distillation objective, carrying out both in a single stage.

For a broader review of unintended text rendering, see App. D.

3 PRELIMINARIES

Flow matching. We use the rectified-flow convention $t \in [0, 1]$, where $t = 1$ is pure noise and $t = 0$ is clean (Lipman et al., 2023; Esser et al., 2024), and denote the condition by c. The linear path is

$$z_t = (1 - t)x_0 + t\epsilon. \tag{1}$$

Here, x_0 is a clean sample and $\epsilon \sim \mathcal{N}(0, I)$, and differentiating Eq. 1 with respect to t gives the target velocity $\mathrm{d}z_t/\mathrm{d}t = \epsilon - x_0$ that the student velocity v_θ regresses onto. We write v_ϕ for the teacher velocity, trained the same way, and use $1 = \sigma_0 > \cdots > \sigma_N = 0$ for the student grid. Sampling advances along that grid by the deterministic Euler update

$$z_{\sigma_{i+1}} = z_{\sigma_i} + (\sigma_{i+1} - \sigma_i)\, v_\theta(z_{\sigma_i}, \sigma_i), \tag{2}$$

whose step $\sigma_{i+1} - \sigma_i$ is negative because the grid decreases from noise to clean.

Distribution matching. At the clean terminal distribution, DMD (Yin et al., 2024b;a) minimizes the reverse KL $\mathrm{D}_{\mathrm{KL}}(q_\theta^{\mathrm{clean}} \,\|\, p_\phi^{\mathrm{clean}})$. TDM (Luo et al., 2025) extends the same marginal matching

to intermediate rollout distributions:

$$\mathcal{L}_{\mathrm{TDM}} = \sum_{i=1}^{N} \mathbb{E}_{\tau\sim\pi_i}\left[\lambda_\tau \, \mathrm{D}_{\mathrm{KL}}(q^i_{\theta,\tau} \,\|\, p_{\phi,\tau})\right]. \tag{3}$$

Here, $q^i_\theta = \mathrm{Law}(z_{\sigma_i})$, $q^i_{\theta,\tau} = k_{\tau|\sigma_i} q^i_\theta$, and $\mathrm{supp}(\pi_i) \subset [\sigma_i, \sigma_{i-1}]$; $p_{\phi,\tau}$ is the teacher marginal and λ_τ is the timestep weight. Its gradient is estimated by

$$\nabla_\theta \mathcal{L}_{\mathrm{TDM}} \approx \sum_{i=1}^{N} \mathbb{E}_{\tau\sim\pi_i,\, z_\tau\sim q^i_{\theta,\tau}}\left[\lambda_\tau J_\tau^\top (s_\psi - s_{\phi,\tau})\right]. \tag{4}$$

Here, $s_\psi \approx s^i_{\theta,\tau} = \nabla_{z_\tau} \log q^i_{\theta,\tau}$, $s_{\phi,\tau} = \nabla_{z_\tau} \log p_{\phi,\tau}$, and $J_\tau = \partial z_\tau / \partial\theta$.

On-policy distillation. OPD (Li et al., 2026; Fang et al., 2026) matches the conditional transition reverse KL at the same student-visited state:

$$\mathrm{D}_{\mathrm{KL}}(q_{\theta,i}(\cdot|z_{\sigma_i}) \,\|\, p_{\phi,i}(\cdot|z_{\sigma_i})) = \omega_i \left\| v_\theta(z_{\sigma_i}, \sigma_i) - v_\phi(z_{\sigma_i}, \sigma_i) \right\|^2. \tag{5}$$

Here, $q_{\theta,i}(\cdot|z_{\sigma_i})$ and $p_{\phi,i}(\cdot|z_{\sigma_i})$ are the student and teacher kernels, $\Delta_i = \sigma_i - \sigma_{i+1} > 0$, and g_i is the diffusion coefficient at σ_i; they share covariance $g_i^2 \Delta_i I$, and $\omega_i = \frac{\Delta_i}{2}(\frac{1}{g_i} + \frac{g_i(1-\sigma_i)}{2\sigma_i})^2$ for $0 < \sigma_i < 1$. The later batch-gated velocity MSE is an unweighted surrogate of this term, and the weight w_b that App. A derives is a separate marginal-KL weight.

4 METHOD

`StepX-OPD` trains one student to carry out multi-teacher on-policy distillation and step distillation at the same time, rather than consolidating capabilities at full step count and compressing the result afterward. We build on trajectory distribution matching (TDM) (Luo et al., 2025) and redraw the rollout grid at random on every training step. Sec. 4.1 then constructs supervision states between the nodes of that grid, and Sec. 4.2 regresses the student's velocity field directly on the low-noise ones. Alg. 1 gives the complete training procedure.

4.1 INTERPOLATED ROLLOUT STATES AND EXACT RE-NOISING

CDM Liu et al. (2026b) replaces TDM's Luo et al. (2025) fixed, predefined discrete anchor schedule with a dynamic time schedule that is redrawn continuously on every training iteration, and `StepX-OPD` adopts this dynamic grid for the rollout it distills. Because the anchors of this grid are now the nodes of the student's own randomized few-step rollout, no state exists at the continuous times that fall between two consecutive nodes, so this section asks how the supervision state at such an in-between time is obtained. `StepX-OPD` builds it in two steps: it interpolates between the two rollout states that bound a sampled segment to obtain a base state, then reaches the supervision state by applying an exact forward re-noising to that base state.

CDM's dynamic schedule also randomizes the number of anchors together with their placement. `StepX-OPD` instantiates the dynamic grid with the segment count fixed and both ends pinned, and randomizes only the placement of the interior nodes. During training, the student advances along a rollout grid $1 = \sigma_0 > \sigma_1 > \cdots > \sigma_N = 0$, and `StepX-OPD` redraws this grid before every training step rather than fixing it to a deterministic schedule. Before any timestep-shift transformation, the underlying uniform grid is divided into N segments. The topmost node remains fixed at $\sigma_0 = 1$, one point is drawn uniformly within every other segment, and the terminal node $\sigma_N = 0$ remains fixed. The grid is redrawn independently at every training step, while all parallel processes share that step's draw so their gradients remain consistent.

Given this randomized rollout grid, `StepX-OPD` first selects one segment $[\sigma_\ell, \sigma_u]$ uniformly at random. It then draws a base time τ_{base} uniformly within that segment, with $\sigma_\ell \le \tau_{\text{base}} < \sigma_u$. Let z_{start} be the state at σ_u and z_{end} the state at σ_ℓ. We interpolate between them using

$$x_{\text{base}} = \lambda \, z_{\text{start}} + (1 - \lambda) \, z_{\text{end}}. \tag{6}$$

Where $\lambda = \frac{\tau_{\text{base}} - \sigma_\ell}{\sigma_u - \sigma_\ell}$. When $\tau_{\text{base}} = \sigma_\ell$, $\lambda = 0$ and Eq. 6 reduces to $x_{\text{base}} = z_{\text{end}}$, so the construction includes direct re-noising from the lower endpoint as a special case. The two rollout states being interpolated are related by Eq. 2, since that is the update the rollout visiting them took, and this relation lets x_{base} be characterized exactly.

Lemma 4.1 (Interpolation is an Euler sub-step)**.** *Suppose the rollout endpoints satisfy the Euler update of Eq. 2 that produced them,* $z_{\mathrm{end}} = z_{\mathrm{start}} + (\sigma_\ell - \sigma_u)\, v_\theta(z_{\mathrm{start}}, \sigma_u)$*. Then the interpolated state of Eq. 6 satisfies*

$$x_{\mathrm{base}} = z_{\mathrm{start}} + (\tau_{\mathrm{base}} - \sigma_u)\, v_\theta(z_{\mathrm{start}}, \sigma_u),$$

so x_{base} *is exactly the state reached by inserting* τ_{base} *into the grid and taking one Euler step from* σ_u*. Moreover, under the sampling independence assumed in App. A, Gaussian re-noising from* x_{base} *to any later time carries* x_{base}*'s own conditional distribution forward exactly, whatever that distribution is, with no approximation.*

Lemma 4.1 identifies x_{base} with the state a first-order Euler step from z_{start} reaches at τ_{base}. This is exactly how CDM constructs its off-trajectory point: it takes an on-trajectory anchor and its predicted velocity and performs a first-order Euler extrapolation toward an independently sampled time. `StepX-OPD`'s within-segment interpolation and CDM's one-step Euler extrapolation therefore compute the same quantity through two different routes, one interpolating between two already-visited rollout states and the other extrapolating from a single velocity evaluation at z_{start}.

Conditioned on the grid and τ_{base}, we sample the supervision time $\tau \sim U[\tau_{\text{base}}, \sigma_u]$, We obtain z_τ from x_{base} by the exact forward SDE transition from τ_{base} to τ, whose coefficients App. A derives by matching moments with the Euler re-noising law of Lemma 4.1. With $\alpha_t = 1 - t$ and $\sigma_t = t$, this re-noising step is

$$z_\tau = \kappa\, x_{\text{base}} + \beta\, \epsilon, \tag{7}$$

where ϵ is a standard Gaussian draw, $\kappa = \frac{\alpha_\tau}{\alpha_{\tau_{\text{base}}}}$, and $\beta = \sqrt{\max\big(\sigma_\tau^2 - (\kappa\, \sigma_{\tau_{\text{base}}})^2,\, 0\big)}$. By Lemma 4.1, this transition carries x_{base}'s own conditional distribution forward to τ exactly, so z_τ has exactly the law obtained by forward-noising the interpolated rollout state itself, with no model prediction standing in for that state. CDM's re-noising step instead starts from the student's own clean-data estimate at its extrapolated point, $\hat{x}_0 = D_\theta(x_{t'_i}, t'_i, c)$, obtained by passing that point back through the student; For the law of the extrapolated state itself, CDM states no exact forward transition from t'_i to its re-noising time. `StepX-OPD` re-noises x_{base} itself, so the exact statement above applies to it directly. The two constructions also sample their supervision time differently. CDM draws its paired time independently of the integration schedule, so its extrapolated point may land beyond the discrete anchors, and because the probability-flow trajectory is curved, a finite Euler stride along the linearized velocity carries that point off the trajectory, which is the off-trajectory supervision its construction is designed to provide. `StepX-OPD` instead keeps τ inside the sampled segment $[\tau_{\text{base}}, \sigma_u]$, so the state being re-noised is always an interpolated rollout state of that segment, which is what Lemma 4.1 requires of it.

Theorem A.2 analyzes how velocity discrepancy appears in a reverse-KL decomposition under its assumptions. Its theoretical weight grows toward the low-noise end of the segment. Motivated by this analysis, we add low-noise velocity MSE as an auxiliary term.

4.2 Low-noise velocity regression

This theorem motivates the auxiliary regression term.

For each sampled state z_τ, an active target model v^{tgt} provides the instantaneous classifier-free guided velocity: during an early warmup it is the frozen initial student $v^{\mathrm{stu}}_{\theta_0}$, and afterward it is the routed specialist $v^{\mathrm{tea}}_{r(d)}$, chosen for source d from M frozen specialists $\{v^{\mathrm{tea}}_m\}_{m=1}^M$ indexed by teacher identity m. It is evaluated once with condition c and once without it at the same state and time:

$$v^{\mathrm{tgt}}_{\mathrm{cfg}}(z_\tau, \tau, c) = v^{\mathrm{tgt}}(z_\tau, \tau, \varnothing) + w\,\big[v^{\mathrm{tgt}}(z_\tau, \tau, c) - v^{\mathrm{tgt}}(z_\tau, \tau, \varnothing)\big]. \tag{8}$$

The trainable student v^{stu}_θ makes an additional forward pass at the same state and time and regresses toward this frozen target through

$$\mathcal{L}_{\mathrm{v}}(\tilde{v}^{\mathrm{tgt}}) = \frac{1}{N_{\mathrm{elt}}} \sum_{j=1}^{B} \big\| v^{\mathrm{stu}}_\theta(z_{\tau_j}, \tau_j, c) - \mathrm{sg}\,\big[\tilde{v}^{\mathrm{tgt}}_j\big] \big\|_2^2, \tag{9}$$

where sg denotes the stop-gradient, B is the number of sampled states in the batch, and N_{elt} is the number of packed tensor elements those states carry. Eq. 9 is the unweighted surrogate of the conditional-transition KL of Eq. 5 anticipated there, and it departs from that KL by dropping the weight ω_i, by averaging the squared distance elementwise rather than summing it, which is where

Table 1: Comparison of text-to-image scores across methods, at forty and at four sampling steps, for both base architectures evaluated. The lower per-architecture block reports the same models after step distillation, and the shaded row is this paper's method.

Base Model	Model	NFEs	GenEval ↑	OCR ↑	DeQA ↑	PickScore ↑	Average ↑
SD3.5-Medium	Base	80	0.666	0.529	4.045	21.79	0.7105
	GenEval Teacher	80	0.946	0.597	3.965	21.66	0.7923
	OCR Teacher	80	0.656	0.924	3.849	21.66	0.7957
	PickScore Teacher	80	0.744	0.629	4.166	23.30	0.7756
	Flow-OPD	80	0.905	0.885	4.268	22.66	0.8788
	DiffusionOPD	80	0.945	0.930	4.113	23.68	0.9022
	Base + TDM	4	0.653	0.335	4.002	21.87	0.6573
	Flow-OPD + TDM	4	0.875	0.751	4.217	22.60	0.8345
	DiffusionOPD + TDM	4	0.838	0.833	3.952	23.41	0.8403
	`StepX-OPD` (Ours)	4	0.944	0.899	4.170	22.98	0.8901
FLUX.1-dev	Base	40	0.655	0.588	4.298	21.96	0.7368
	GenEval Teacher	40	0.900	0.663	4.333	22.06	0.8195
	OCR Teacher	40	0.670	0.942	4.254	21.99	0.8270
	PickScore Teacher	40	0.690	0.678	4.153	23.57	0.7765
	Flow-OPD	40	0.880	0.913	4.247	23.50	0.8864
	DiffusionOPD	40	0.892	0.933	4.225	23.58	0.8943
	Base + TDM	4	0.632	0.557	4.340	21.94	0.7252
	Flow-OPD + TDM	4	0.864	0.901	4.087	23.30	0.8695
	DiffusionOPD + TDM	4	0.882	0.925	4.005	23.32	0.8761
	`StepX-OPD` (Ours)	4	0.890	0.929	4.233	23.28	0.8902

the $1/N_{\text{elt}}$ factor comes from, and by taking as its target the guided, per-step $\tilde{v}^{\text{tgt}} = v^{\text{tgt}}_{\text{cfg}}(z_\tau, \tau, c)$ of Eq. 8 instead of a single fixed teacher velocity v_ϕ. The auxiliary term is gated for the whole batch: with training step s, configured start step s_{v}, sampled times $\{\tau_j\}_{j=1}^{B}$, and threshold τ_{v},

$$g_{\text{v}} = \mathbf{1}\{s \geq s_{\text{v}}\}\ \mathbf{1}\{\bar{\tau} \leq \tau_{\text{v}}\}, \qquad \bar{\tau} = \frac{1}{B}\sum_{j=1}^{B} \tau_j. \tag{10}$$

When $g_{\text{v}} = 1$, the velocity loss applies to the entire batch; otherwise it contributes zero. This supplies direct local velocity supervision near the clean endpoint, where the capability signal must be transferred through the student's on-policy states. The main-run coefficient w_{v}, threshold, and start step are given in App. B.

The same target $\tilde{v}^{\text{tgt}}$ also supplies the teacher score for the distribution-matching term $\mathcal{L}_{\text{TDM}}(\tilde{v}^{\text{tgt}})$, which follows the gradient estimator of Eq. 4 with the teacher score $s_{\phi,\tau}$ evaluated at $\tilde{v}^{\text{tgt}}$ rather than at a single fixed teacher. App. B gives the differentiable surrogate that turns this gradient into a loss through the endpoint state z_{end}. The total objective is

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{TDM}}(\tilde{v}^{\text{tgt}}) + w_{\text{v}} g_{\text{v}}\, \mathcal{L}_{\text{v}}(\tilde{v}^{\text{tgt}}), \tag{11}$$

where $\tilde{v}^{\text{tgt}}$ is the instantaneous guided target of Eq. 8 in both terms. App. B gives the main-run values of w_{v}, the start step, and the batch-mean threshold for velocity regression, along with Alg. 1, which summarizes one training iteration of this procedure. See App. G for an optional target that replaces this instantaneous velocity with an average taken along a short guided rollout from the supervision state.

5 Experiments

5.1 Experimental setup

Model. We evaluate two base architectures, SD3.5-Medium (Esser et al., 2024) and FLUX.1-dev (Labs et al., 2025). For each, the joint OPD models and task-specialist teachers use forty-step generation and the step-distilled students, including `StepX-OPD`, use four steps. See App. B for the sampling resolution and guidance scale used at each step count.



Figure 2: Qualitative comparison result from benchmark.

Benchmark. We evaluate each base model, its task specialists, and its joint OPD models at forty steps, together with step-distilled variants of the base and joint OPD models at four steps, on GenEval (Ghosh et al., 2023), OCR accuracy (Cui et al., 2025), DeQA (You et al., 2025), and PickScore (Kirstain et al., 2023), combined into a single Average score. Tab. 1 reports these four measures and Average for both SD3.5-Medium and FLUX.1-dev under the same protocol, and Tab. 7 additionally reports CLIPScore (Xu et al., 2024), HPSv2.1 (Wu et al., 2023), Aesthetic (Murray et al., 2012), ImageReward (Xu et al., 2023), and UnifiedReward (Wang et al., 2025) for the SD3.5-Medium rows, outside the Average score. See App. C for the Average formula, the generated image counts, and the scoring procedures.

Implementation details. For the four-step distillation runs, we adapt the transformer with rank-32 LoRA student and fake adapters, trained at learning rates 2×10^{-6} and 1×10^{-5} respectively, on four-step on-policy rollouts. Multi-teacher supervision covers the GenEval, OCR, and PickScore training sources, and the reported configuration combines trajectory distribution matching with low-noise velocity regression against the routed teacher under static source routing. See App. B for the LoRA initialization, batch size, warmup, and teacher-routing schedule.

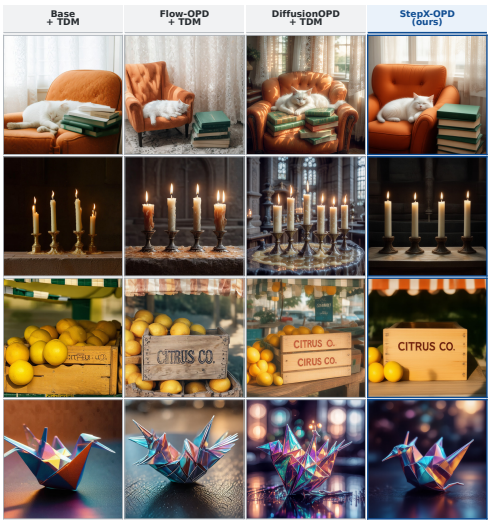


Figure 3: Qualitative comparison result from generated prompt.

5.2 Main Results

Quantitative. To test whether joint training can deliver integrated capability at four steps, Tab. 1 compares `StepX-OPD` with four-step TDM baselines and forty-step OPD models on both SD3.5-Medium and FLUX.1-dev. `StepX-OPD` reaches a four-step Average of 0.8901 on SD3.5-Medium and 0.8902 on FLUX.1-dev, against 0.8403 and 0.8761 for the next-best four-step baseline on each. This shows that joint capability consolidation and trajectory compression can deliver integrated capability at four steps.

Qualitative comparison. Fig. 2 compares the forty-step and four-step models on one prompt from each of two benchmarks, and shows `StepX-OPD` matching what the forty-step models produce while the other four-step columns fall short of both. It is the only column of the six that spells the

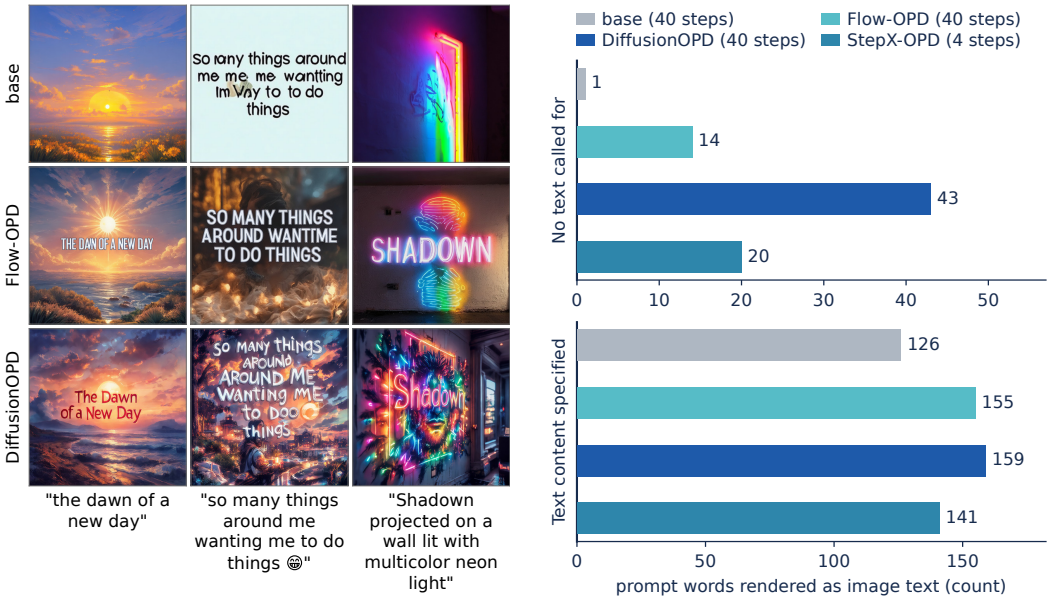


Figure 4: Text that models render into the image. The left panel shows generations on the *No text called for* prompt group; the right panel counts the generations whose detected on-image text reproduces a word from the prompt, on that same group and on *Text content specified*, drawn on separate scales. See App. C for the exact prompt selection of both groups.

poster title correctly. Fig. 3 compares the four-step models on prompts written for this comparison rather than taken from a benchmark. Further examples appear in App. F.

5.3 Prompt-word recurrence

To test whether capability integration is accompanied by unintended prompt-word rendering, Tab. 12 counts recurrence across eight models on prompts that do not request text, with seven models sampled at forty steps and `StepX-OPD` at four steps; Fig. 4 repeats the comparison on a stricter prompt subset. The largest observed shift is on SD3.5-Medium, where recurrence rises from 21 images for the base model to 218 for DiffusionOPD. This shows that the recurrence measure exposes a text-rendering side effect associated with capability integration.

5.4 Ablation and analysis

Supervision-point construction. To assess supervision-point construction, Tab. 2 compares fixed-grid endpoints, randomized-grid endpoints, and randomized-grid interpolated re-noising while holding every other factor fixed: all three rows use static source routing to the same three Flow-GRPO expert teachers on the GenEval, OCR, and PickScore training sources, start the student LoRA from the same base initialization, and train at the same per-rank batch size across 4 GPUs without low-noise velocity regression. Replacing the fixed grid with the randomized one raises Average from 0.7090 to 0.7548, and moving the re-noising source from the segment endpoint to an interpolated point raises it further to 0.8240, with OCR rising from 0.4499 to 0.5776 to 0.7968 across the same three rows. Both changes therefore improve the aggregate score in a comparison that isolates supervision-point construction from teacher family and initialization.

Table 2: Ablation of the three supervision-point constructions, best results are in bold.

Rollout grid	Re-noising source	GenEval ↑	OCR ↑	DeQA ↑	PickScore ↑	Average ↑
Fixed	Segment endpoint	0.7230	0.4499	4.003	22.42	0.7090
Randomized	Segment endpoint	0.7523	0.5776	**4.129**	22.46	0.7548
Randomized	Interpolated	**0.8198**	**0.7968**	4.076	**22.47**	**0.8240**

Velocity-regression target source. To examine the velocity-regression target source, Tab. 3 compares routed and dedicated targets under low-noise coverage. The routed row reaches an Average of 0.8901, versus 0.8446 for the dedicated-target row. Because the rows also differ in initialization, teacher family, and target construction, this observed ordering only shows a higher aggregate score for the reported routed variant and does not isolate a causal effect of target source.

Initialization from released adapters. To test dependence on student initialization, Tab. 4 compares TDM and `StepX-OPD` from the released Flow-OPD and DiffusionOPD adapters. From the Flow-OPD adapter, `StepX-OPD` raises Average from 0.8345 to 0.8746, the larger gain of the two pairs. The higher measured ordering from both adapters shows that the advantage does not depend on the additive-merge initialization, while the data-source difference remains a caveat.

Table 3: Ablation of velocity-regression noise coverage crossed with the model that supplies the regression target. Best results are in bold.

Coverage	Target	Gen ↑	OCR ↑	DeQA ↑	Pick ↑	Avg ↑
Low-noise	Routed	**0.9442**	**0.8986**	**4.170**	**22.98**	**0.8901**
	Dedicated	0.8822	0.8127	4.057	22.67	0.8446
Full-range	Routed	0.8894	0.8341	4.121	22.73	0.8555
	Dedicated	0.8664	0.7296	4.101	22.65	0.8219

Table 4: Comparison of distillation methods across two student initializations. Best results are in bold.

Init.	Method	Gen ↑	OCR ↑	DeQA ↑	Pick ↑	Avg ↑
Flow-OPD	TDM	0.875	0.751	**4.217**	22.60	0.8345
	`StepX-OPD`	**0.909**	**0.897**	4.103	**22.68**	**0.8746**
DiffusionOPD	TDM	0.838	0.833	3.952	**23.41**	0.8403
	`StepX-OPD`	**0.906**	**0.893**	**4.056**	22.92	**0.8727**

Velocity-regression starting step. To examine when velocity regression should begin, Tab. 5 compares starts at steps 500 and 1000. Starting at step 500 reaches an Average of 0.8901, versus 0.8426 from step 1000. Thus, the earlier start is associated with stronger aggregate performance in the reported runs.

SFT-increment merge weight. To compare how the SFT increment enters the student initialization, Tab. 6 tests the base merge, a four-model mean, and an additive merge. The additive variant reaches the highest Average, 0.8348 versus 0.8219 for the next-best row. This supports the additive merge as the strongest reported initialization variant.

Table 5: Ablation of the step at which the velocity-regression term starts, reported as full-benchmark scores. Best results are in bold.

Start	Gen ↑	OCR ↑	DeQA ↑	Pick ↑	Avg ↑
step 500	**0.9442**	**0.8986**	**4.170**	**22.98**	**0.8901**
step 1000	0.8869	0.8145	3.996	22.62	0.8426

Table 6: Ablation of SFT-merge variants of the student initialization, reported as full-benchmark scores. Best results are in bold.

Initialization	Gen ↑	OCR ↑	DeQA ↑	Pick ↑	Avg ↑
Merge	0.8664	**0.7296**	4.101	22.65	0.8219
SFT Merge (mean)	0.8758	0.7054	**4.116**	22.65	0.8189
SFT Merge (additive)	**0.9273**	0.7294	4.050	**22.68**	**0.8348**

6 Conclusion

In this paper, we introduced `StepX-OPD` to jointly consolidate reward-specialized text-to-image capabilities and compress generation to four steps. This avoids using a consolidated full-step model as a later compression teacher. Furthermore, source-routed teachers supervise the student's own randomized few-step rollouts. Within-segment interpolation and exact forward re-noising align capability transfer with the trajectory being compressed. Our theoretical analysis motivates low-noise vMSE, which adds direct local supervision from the routed teacher near the clean end. The resulting method is a single-stage, trajectory-aligned, multi-teacher on-policy distillation framework for capability consolidation and few-step compression. Extensive experiments on SD3.5-Medium show that `StepX-OPD` reaches an Average of 0.8901 in four sampling steps, compared with 0.8403 for the next-best four-step baseline.

LLM USE STATEMENT

We used large language model tools to polish the writing in this manuscript, including improving wording and phrasing during editing. The authors reviewed all AI-assisted text before it was included in the paper. We take responsibility for the final content of this work, including its text, claims, and artifacts.

ETHICS STATEMENT

This work does not involve human subjects, and it does not collect new human annotations. The evaluations in this paper use public benchmarks, and the base and comparison models draw on publicly released weights. `StepX-OPD` does not introduce a new generative capability: it compresses the sampling trajectory of an existing diffusion model and consolidates capabilities from existing reward-specialized teachers, so it inherits the risks already associated with its base model, such as generating misleading or harmful imagery, rather than introducing a new category of risk. One consideration follows directly from our results: `StepX-OPD` improves the rendering of text inside generated images, and this improvement lowers the barrier to fabricating images that contain text, such as forged signage, storefronts, or official documents. Reducing the number of sampling steps also lowers the computational cost of inference, which reduces the energy required per generated image but likewise lowers the barrier to generating images at scale.

REPRODUCIBILITY STATEMENT

Sec. 4 specifies the continuous-time construction, teacher targets, and velocity-regression objective. Sec. 5 defines the compared model families, metrics, and normalized Average, while App. B and App. C record the resolved training configuration and scoring protocol. Together these details identify the reported settings and their evaluation records.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/5be69a584901a26c521c2b51e40a4c20-Paper-Conference.pdf.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *International Conference on Learning Representations*, volume 2024, pp. 4965–4987, 2024.

Jingye Chen, Yupan Huang, Tengchao Lv, Lei Cui, Qifeng Chen, and Furu Wei. Textdiffuser: Diffusion models as text painters. *Advances in Neural Information Processing Systems*, 36:9353–9387, 2023.

Cheng Cui, Ting Sun, Manhui Lin, Tingquan Gao, Yubo Zhang, Jiaxuan Liu, Xueqing Wang, Zelun Zhang, Changda Zhou, Hongen Liu, Yue Zhang, Wenyu Lv, Kui Huang, Yichao Zhang, Jing Zhang, Jun Zhang, Yi Liu, Dianhai Yu, and Yanjun Ma. Paddleocr 3.0 technical report, 2025. URL https://arxiv.org/abs/2507.05595.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Zhen Fang, Wenxuan Huang, Yu Zeng, Yiming Zhao, Shuang Chen, Kaituo Feng, Yunlong Lin, Lin Chen, Zehui Chen, Shaosheng Cao, et al. Flow-opd: On-policy distillation for flow matching models. *arXiv preprint arXiv:2605.08063*, 2026.

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. *arXiv preprint arXiv:2506.00523*, 2025.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: an object-focused framework for evaluating text-to-image alignment. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Yunqi Hong, Kuei-Chun Kao, Hengguang Zhou, and Cho-Jui Hsieh. Understanding reward hacking in text-to-image reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 4245–4255, June 2026.

Haoqi Hu, Tongji Luo, Li Zhang, and Boning Zhou. Tangpoetrybench: A multi-dimensional benchmark and rubric-conditioned evaluator for poetry-to-image generation. *arXiv preprint arXiv:2608.11452*, 2026.

Dengyang Jiang, Xin Jin, Dongyang Liu, Zanyi Wang, Mingzhe Zheng, Ruoyi Du, Xiangpeng Yang, Qilong Wu, Zhen Li, Peng Gao, Harry Yang, and Steven Hoi. D-opsd: On-policy self-distillation for continuously tuning step-distilled diffusion models. *arXiv preprint arXiv:2605.05204*, 2026.

Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: an open dataset of user preferences for text-to-image generation. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space, 2025. URL https://arxiv.org/abs/2506.15742.

Quanhao Li, Junqiu Yu, Kaixun Jiang, Yujie Wei, Zhen Xing, Pandeng Li, Ruihang Chu, Shiwei Zhang, Yu Liu, and Zuxuan Wu. Diffusionopd: A unified perspective of on-policy distillation in diffusion models. *SIGGRAPH Asia 2026 Conference Papers*, 2026.

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

Dongyang Liu, Peng Gao, David Liu, Ruoyi Du, Zhen Li, Qilong Wu, Xin Jin, Sihan Cao, Shifeng Zhang, Steven HOI, and Hongsheng Li. Decoupled DMD: CFG augmentation as the spear, distribution matching as the shield. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=jBztvOiCKE.

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 *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=oCBKGw5HNf.

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, 2026b. URL https://arxiv.org/abs/2605.06376.

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

Siwei Lyu. Interpretation and generalization of score matching. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pp. 359–366, Arlington, Virginia, USA, 2009. AUAI Press. ISBN 9780974903958.

Joanna Materzyńska, Antonio Torralba, and David Bau. Disentangling visual and written concepts in clip. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16389–16398, 2022. doi: 10.1109/CVPR52688.2022.01592.

Naila Murray, Luca Marchesotti, and Florent Perronnin. Ava: A large-scale database for aesthetic visual analysis. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2408–2415, 2012. doi: 10.1109/CVPR.2012.6247954.

Juan Rodriguez, Haotian Zhang, Abhay Puri, Rishav Pramanik, Aarash Feizi, Pascal Wichmann, Arnab Mondal, Mohammad R Samsami, Rabiul Awal, Perouz Taslakian, et al. Rendering-aware reinforcement learning for vector graphics generation. In *NeurIPS*, 2025.

Tim Salimans, Thomas Mensink, Jonathan Heek, and Emiel Hoogeboom. Multistep distillation of diffusion models via moment matching. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

Yibin Wang, Yuhang Zang, Hao Li, Cheng Jin, and Jiaqi Wang. Unified reward model for multimodal understanding and generation. *arXiv preprint arXiv:2503.05236*, 2025.

Tianhe Wu, Kun Yan, Zikai Zhou, Lihan Jiang, Jiahao Li, Jie Zhang, Kaiyuan Gao, Ningyuan Tang, Shengming Yin, Xiaoyue Chen, et al. Qwen-image-flash: Beyond objective design. *arXiv preprint arXiv:2606.03746*, 2026.

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.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: learning and evaluating human preferences for text-to-image generation. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pp. 15903–15935, 2023.

Zipeng Xu, Songlong Xing, Enver Sangineto, and Nicu Sebe. Spectralclip: Preventing artifacts in text-guided style transfer from a spectral perspective. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 5109–5118, 2024. doi: 10.1109/WACV57701.2024.00504.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Frédo Durand, and William T. Freeman. Improved distribution matching distillation for fast image synthesis. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 47455–47487. Curran Associates, Inc., 2024a. doi: 10.52202/079017-1505. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/54dcf25318f9de5a7a01f0a4125c541e-Paper-Conference.pdf.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6613–6623, 2024b. doi: 10.1109/CVPR52733.2024.00632.

TaeHo Yoon, Kibeom Myoung, Keon Lee, Jaewoong Cho, Albert No, and Ernest K. Ryu. Censored sampling of diffusion models using 3 minutes of human feedback. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Zhiyuan You, Xin Cai, Jinjin Gu, Tianfan Xue, and Chao Dong. Teaching large language models to regress accurate image quality scores using score distribution. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14483–14494, 2025.

Mingyuan Zhou, Huangjie Zheng, Zhendong Wang, Mingzhang Yin, and Hai Huang. Score identity distillation: exponentially fast distillation of pretrained diffusion models for one-step generation. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Wei Zhou, Xiongwei Zhu, Zelin Xu, Bo Dong, Lixue Gong, Yongyuan Liang, Meng Chu, Leigang Qu, Lingdong Kong, Wei Liu, et al. Danceopd: On-policy generative field distillation. *arXiv preprint arXiv:2606.27377*, 2026.

APPENDIX

CONTENTS OF APPENDIX

A Theoretical Analysis **15**

A.1 Setting and Assumptions 15

A.2 Preliminary Identities 16

A.3 Main Results and Proofs 17

A.4 Training Objectives and Low-Noise Truncation 20

B Implementation details for the reported static-routing variant **20**

C Evaluation protocol **22**

D Additional related work on unintended text rendering **22**

E Supplementary results **23**

F Additional qualitative comparisons **30**

G Precise teacher targets and prompt-word recurrence **34**

H Soup composition and velocity-regression variants **35**

A Theoretical Analysis

A.1 Setting and Assumptions

DMD matches distributions after forward noising, and TDM applies this objective to states along the student's trajectory (Yin et al., 2024b; Luo et al., 2025). OPD matches the teacher's behavior at states generated by the student (Li et al., 2026). We fix a text condition and relate these objectives on the same noised states.

A.1.1 Sampling and Forward Laws

Given a random rollout grid, first select one of its K segments uniformly at random and then draw the base time uniformly within that segment:

$$\mathcal{T} = (1 = t_K > \cdots > t_0 = 0), \qquad i \sim \mathrm{Unif}\{1, \ldots, K\}, \qquad b \mid i \sim \mathrm{Unif}[t_{i-1}, t_i). \tag{12}$$

Interpolate the two endpoint rollout states to obtain the state at the sampled base:

$$X_b = \frac{(t_i - b)X_{t_{i-1}} + (b - t_{i-1})X_{t_i}}{t_i - t_{i-1}}. \tag{13}$$

The analysis below conditions on the sampled $(\mathcal{T}, b)$, with $b < t_i \leq 1$. Segment lengths vary under a random grid, so this two-stage draw does not make b itself uniform on $(0, 1)$; conditioning on $(\mathcal{T}, b)$ means this marginal law does not affect the conclusions that follow. When $t_i = 1$, the uniform draw satisfies $T < 1$ almost surely, so every velocity and identity is evaluated below 1. Values assigned at the sampling endpoints do not affect the Lebesgue expectations. Let q_b be the conditional distribution of X_b, and let p_b be the teacher distribution at the same time. All laws are on $\mathbb{R}^d$, and $D_{\mathrm{KL}}(q\|p)$ denotes relative entropy. For $b < t < 1$, apply the same forward noising kernel to both laws:

$$\begin{aligned} k_{t|b}(\cdot \mid x) &= \mathcal{N}(a_{t|b}x, s_{t|b}^2 I), \\ a_{t|b} = \frac{1-t}{1-b}, &\qquad s_{t|b}^2 = t^2 - a_{t|b}^2 b^2. \end{aligned} \tag{14}$$

$$q_t = k_{t|b} q_b, \qquad p_t = k_{t|b} p_b. \tag{15}$$

Here $kr = \int k(\cdot \mid x) r(\mathrm{d}x)$, and q_t is the forward-noised law from q_b. Next sample the supervision time and apply the exact forward SDE transition from b to T:

$$\begin{aligned} T \mid \mathcal{T}, b \sim \mathrm{Unif}[b, t_i], &\qquad \epsilon \sim \mathcal{N}(0, I), \\ Z_T = a_{T|b}X_b + s_{T|b}\epsilon, &\qquad Z_T \mid T, \mathcal{T}, b \sim q_T. \end{aligned} \tag{16}$$

Unless stated otherwise, expectations are conditional on $(\mathcal{T}, b)$. The weight used throughout is

$$w_b(t) = (t_i - t)\frac{1-t}{t}, \qquad b < t < t_i. \tag{17}$$

A.1.2 Assumptions

The statements below use the following assumptions at the fixed grid and base.

- **A1. Conditional sampling and common kernel.** Given $(\mathcal{T}, b)$, T is independent of X_b, and ϵ is independent of (X_b, T). Both distributions evolve through the kernel in equation 14, and the samples follow equation 16.
- **A2. Regularity and the base limit.** The laws q_b, p_b have finite second moments and $D_{\mathrm{KL}}(q_b\|p_b) < \infty$. Their Gaussian-smoothed densities satisfy the smoothness and tail conditions of the relative de Bruijn identity below. In particular, differentiation under the integral and integration by parts are valid, with vanishing boundary terms. The function $K(t) = D_{\mathrm{KL}}(q_t\|p_t)$ is continuously differentiable on (b, t_i) and satisfies $K(t) \to D_{\mathrm{KL}}(q_b\|p_b)$ as $t \downarrow b$.
- **A3. Teacher consistency.** Let v_ϕ denote the teacher network velocity. Assume $v_\phi(z, t) = v_p(z, t)$ for $b < t < t_i$, where v_p is the distribution-induced velocity in equation 27. The student network velocity is the measurable function v_θ. The teacher network velocity v_ϕ is the velocity field produced by the teacher together with its guidance configuration, and p is defined to be the distribution induced by this guided field, following the convention used throughout distillation methods built on classifier-free guidance.
- **A4. Weighted integrability.** The weighted squared discrepancies defining O_b and R_b in equation 35 and equation 36 are integrable.

A.2 Preliminary Identities

A.2.1 Semigroup of the Forward Kernel

For $b < s < t < 1$, the Gaussian kernels satisfy

$$k_{t|s}k_{s|b} = k_{t|b}. \tag{18}$$

Indeed, composition gives a Gaussian kernel, whose mean coefficient and variance are

$$a_{t|s}a_{s|b} = \frac{1-t}{1-s}\frac{1-s}{1-b} = a_{t|b}, \tag{19}$$

$$\begin{aligned} a_{t|s}^2 s_{s|b}^2 + s_{t|s}^2 &= a_{t|s}^2\big(s^2 - a_{s|b}^2 b^2\big) + t^2 - a_{t|s}^2 s^2 \\ &= t^2 - a_{t|b}^2 b^2 = s_{t|b}^2. \end{aligned} \tag{20}$$

Consequently, $q_t = k_{t|s}q_s$ and $p_t = k_{t|s}p_s$. Both smoothed laws therefore follow the same forward evolution.

A.2.2 Euler Interpolation and Exact Re-noising

Proof of Lemma 4.1. Fix the sampled grid and base. By hypothesis, the two rollout endpoints satisfy the Euler update

$$X_{t_{i-1}} = X_{t_i} + (t_{i-1} - t_i)v_\theta(X_{t_i}, t_i). \tag{21}$$

Substitute equation 21 into equation 13 and collect the coefficients of X_{t_i}:

$$\begin{aligned} X_b &= \frac{(t_i - b)\big[X_{t_i} + (t_{i-1} - t_i)v_\theta(X_{t_i}, t_i)\big] + (b - t_{i-1})X_{t_i}}{t_i - t_{i-1}} \\ &= X_{t_i} + \frac{(t_i - b)(t_{i-1} - t_i)}{t_i - t_{i-1}} v_\theta(X_{t_i}, t_i) \\ &= X_{t_i} + (b - t_i)v_\theta(X_{t_i}, t_i). \end{aligned} \tag{22}$$

The sum of those coefficients is $[(t_i{-}b){+}(b{-}t_{i-1})]/(t_i{-}t_{i-1}) = 1$, and $(t_{i-1}{-}t_i)/(t_i{-}t_{i-1}) = -1$ gives the last line, so equation 22 is exactly the state reached by inserting b into the grid and advancing from t_i to b by one Euler step. Inserting b below t_i leaves the already computed states at $t_K, \dots, t_i$ unchanged. When $b = t_{i-1}$, equation 21 gives the lower endpoint directly, matching equation 22 at that boundary. Under the conditional independence assumed in Assumption A1, re-noising from this base has exactly the law $k_{t|b}q_b$ for $b < t < 1$, where q_b is the actual conditional law of the interpolated state. For re-noising, condition on $X_b = x$. Independence of ϵ makes $Z_t = a_{t|b}x + s_{t|b}\epsilon$ Gaussian with kernel $k_{t|b}(\cdot \mid x)$. Thus, for every Borel set A,

$$\begin{aligned} \mathbb{P}(Z_t \in A \mid \mathcal{T}, b) &= \int \mathbb{P}(Z_t \in A \mid X_b = x, \mathcal{T}, b)\, q_b(\mathrm{d}x) \\ &= \int k_{t|b}(A \mid x)\, q_b(\mathrm{d}x) = (k_{t|b}q_b)(A). \end{aligned} \tag{23}$$

For $b < s < t < 1$, the Chapman–Kolmogorov identity equation 18 also gives

$$k_{t|s}(k_{s|b}q_b) = (k_{t|s}k_{s|b})q_b = k_{t|b}q_b. \tag{24}$$

so adding noise in two stages and adding it directly give the same marginal law. □

Relation to the continuous ODE. The interpolation identity is exact for the piecewise-linear Euler trajectory. For a continuous solution of $\dot{x}(t) = v_\theta(x(t), t)$, the Euler sub-step is a numerical approximation. Even when both endpoints lie on the same exact ODE solution, linear interpolation reproduces every intermediate state exactly only when that solution segment is affine, equivalently when its velocity is constant along the segment. Indeed, agreement with the endpoint chord at every intermediate time makes the trajectory affine, and the converse follows by interpolation. The Gaussian transition above is exact for the actual base law q_b, including any approximation error in that law relative to a continuous ODE marginal.

Coefficients of the re-noising step used in the main text. Sec. 4 moves from τ_{base} to τ along the forward path $z_t = \alpha_t x_0 + \sigma_t \epsilon$ with $\alpha_t = 1 - t$ and $\sigma_t = t$, writing this transition as $z_\tau = \kappa\, x_{\text{base}} + \beta\, \epsilon$ in Eq. 7. This step is the kernel of Eq. 14 evaluated at $t = \tau$ and $b = \tau_{\text{base}}$, so $\kappa = a_{\tau|\tau_{\text{base}}}$ and $\beta = s_{\tau|\tau_{\text{base}}}$, and the derivation below writes the two coefficients out directly in the α_t, σ_t notation that Sec. 4 uses. Requiring $z_\tau = \kappa\, x_{\text{base}} + \beta\, \epsilon'$, with ϵ' independent of the noise already carried by

x_{base}, to share the same law as $\alpha_\tau x_0 + \sigma_\tau \epsilon$ fixes both coefficients by matching moments. Matching the coefficient of x_0 gives $\kappa\, \alpha_{\tau_{\text{base}}} = \alpha_\tau$, hence

$$\kappa = \frac{\alpha_\tau}{\alpha_{\tau_{\text{base}}}}, \tag{25}$$

and matching the variance contributed by the Gaussian noise gives $(\kappa\, \sigma_{\tau_{\text{base}}})^2 + \beta^2 = \sigma_\tau^2$, hence

$$\beta^2 = \sigma_\tau^2 - (\kappa\, \sigma_{\tau_{\text{base}}})^2. \tag{26}$$

Under the ordering $\tau \geq \tau_{\text{base}}$ that Eq. 7 assumes, this right-hand side is nonnegative, so the $\max(\cdot, 0)$ that Eq. 7 applies before the square root only guards against floating-point round-off at that boundary, not against a genuine sign change.

A.2.3 Velocity and Score

For $r \in \{q, p\}$, define the velocity induced by the forward law:

$$v_r(z,t) = -\frac{z}{1-t} - \frac{t}{1-t}\nabla_z \log r_t(z). \tag{27}$$

Subtracting the two definitions cancels the common drift. Thus,

$$v_q(z,t) - v_p(z,t) = -\frac{t}{1-t}\big(\nabla_z \log q_t(z) - \nabla_z \log p_t(z)\big), \tag{28}$$

$$\|\nabla_z \log q_t(z) - \nabla_z \log p_t(z)\|^2 = \frac{(1-t)^2}{t^2}\|v_q(z,t) - v_p(z,t)\|^2. \tag{29}$$

A.2.4 Relative de Bruijn Identity

Let $u > 0$ be a Gaussian variance parameter and define

$$\widehat{q}_u = q_b * \mathcal{N}(0, uI), \qquad \widehat{p}_u = p_b * \mathcal{N}(0, uI). \tag{30}$$

Theorem 1 of Lyu (2009) gives the relative KL dissipation identity under Gaussian smoothing, also called a generalized de Bruijn identity:

$$\frac{\mathrm{d}}{\mathrm{d}u} D_{\mathrm{KL}}(\widehat{q}_u \| \widehat{p}_u) = -\frac{1}{2}\mathbb{E}_{Y\sim\widehat{q}_u}\|\nabla_y \log \widehat{q}_u(Y) - \nabla_y \log \widehat{p}_u(Y)\|^2. \tag{31}$$

We use this established identity directly. The change of variables for our forward kernel is given in Proposition A.1.

A.2.5 A Uniform Random Integration Endpoint

For any measurable $h \geq 0$ and $T \sim \mathrm{Unif}[b, t_i]$, Tonelli's theorem gives

$$\begin{aligned}
\mathbb{E}_T\left[\int_b^T h(t)\,\mathrm{d}t\right] &= \frac{1}{t_i - b}\int_b^{t_i}\int_b^{\tau} h(t)\,\mathrm{d}t\,\mathrm{d}\tau \\
&= \frac{1}{t_i - b}\int_b^{t_i}\int_t^{t_i} h(t)\,\mathrm{d}\tau\,\mathrm{d}t \\
&= \frac{1}{t_i - b}\int_b^{t_i}(t_i - t)h(t)\,\mathrm{d}t \\
&= \mathbb{E}_T\big[(t_i - T)h(T)\big].
\end{aligned} \tag{32}$$

The factor $1/(t_i - b)$ is already included in the uniform expectation in the last line.

A.3 Main Results and Proofs

For an Euler rollout, Lemma 4.1 identifies q_b as the law of an interpolated Euler state and justifies its exact Gaussian smoothing. The KL results below apply to this actual base law under the stated assumptions. Define

$$K(t) = D_{\mathrm{KL}}(q_t \| p_t), \qquad m(t) = \mathbb{E}_{Z\sim q_t}\|v_q(Z,t) - v_p(Z,t)\|^2. \tag{33}$$

The three quantities used in the main theorem are

$$D_b = \mathbb{E}_T\, D_{\mathrm{KL}}(q_T \| p_T), \tag{34}$$

$$O_b = \mathbb{E}\big[w_b(T)\|v_\theta(Z_T,T) - v_\phi(Z_T,T)\|^2\big], \tag{35}$$

$$R_b = \mathbb{E}\big[w_b(T)\|v_\theta(Z_T,T) - v_q(Z_T,T)\|^2\big]. \tag{36}$$

Proposition A.1 (Dissipation under the forward kernel)**.** *Under Assumptions A1 and A2, for $b < t < t_i$,*

$$K'(t) = -\frac{1-t}{t} m(t). \tag{37}$$

Proof. For this proof, write $a(t) = a_{t|b}$. The rescaled state and its Gaussian noise variance are

$$\begin{aligned} Y_t &= \frac{Z_t}{a(t)} = X_b + \sqrt{\lambda_b(t)}\,\epsilon, \\ \lambda_b(t) &= \frac{s_{t|b}^2}{a(t)^2} = \frac{(1-b)^2 t^2}{(1-t)^2} - b^2. \end{aligned} \tag{38}$$

The kernel in equation 14 is equivalent to Gaussian smoothing with variance $\lambda_b(t)$ followed by multiplication by $a(t)$. The resulting densities are

$$q_t(z) = a(t)^{-d} \widehat{q}_{\lambda_b(t)}(z/a(t)), \qquad p_t(z) = a(t)^{-d} \widehat{p}_{\lambda_b(t)}(z/a(t)). \tag{39}$$

Using the substitution $y = z/a(t)$ and $\mathrm{d}z = a(t)^d \mathrm{d}y$, the common scaling cancels in KL:

$$\begin{aligned} K(t) &= \int a(t)^{-d} \widehat{q}_{\lambda_b(t)}(z/a(t)) \log \frac{\widehat{q}_{\lambda_b(t)}(z/a(t))}{\widehat{p}_{\lambda_b(t)}(z/a(t))} \,\mathrm{d}z \\ &= \int \widehat{q}_{\lambda_b(t)}(y) \log \frac{\widehat{q}_{\lambda_b(t)}(y)}{\widehat{p}_{\lambda_b(t)}(y)} \,\mathrm{d}y \\ &= D_{\mathrm{KL}}(\widehat{q}_{\lambda_b(t)} \| \widehat{p}_{\lambda_b(t)}). \end{aligned} \tag{40}$$

The spatial chain rule applied to equation 39 gives, for $r \in \{q, p\}$,

$$\nabla_z \log r_t(z) = a(t)^{-1} \nabla_y \log \widehat{r}_{\lambda_b(t)}(z/a(t)). \tag{41}$$

Since $Z = a(t)Y$ has law q_t when $Y \sim \widehat{q}_{\lambda_b(t)}$, this implies

$$\begin{aligned} &\mathbb{E}_{Y \sim \widehat{q}_{\lambda_b(t)}} \|\nabla \log \widehat{q}_{\lambda_b(t)}(Y) - \nabla \log \widehat{p}_{\lambda_b(t)}(Y)\|^2 \\ &\quad = a(t)^2 \mathbb{E}_{Z \sim q_t} \|\nabla \log q_t(Z) - \nabla \log p_t(Z)\|^2. \end{aligned} \tag{42}$$

Differentiating the variance parameter in equation 38 gives

$$\begin{aligned} \lambda_b'(t) &= (1-b)^2 \left(\frac{2t}{(1-t)^2} + \frac{2t^2}{(1-t)^3} \right) \\ &= \frac{2(1-b)^2 t}{(1-t)^3}. \end{aligned} \tag{43}$$

Apply the time chain rule to equation 40, substitute Lyu's identity equation 31, and then use equation 42:

$$\begin{aligned} K'(t) &= \lambda_b'(t) \left. \frac{\mathrm{d}}{\mathrm{d}u} D_{\mathrm{KL}}(\widehat{q}_u \| \widehat{p}_u) \right|_{u=\lambda_b(t)} \\ &= -\frac{\lambda_b'(t)}{2} \mathbb{E}_{Y \sim \widehat{q}_{\lambda_b(t)}} \|\nabla \log \widehat{q}_{\lambda_b(t)}(Y) - \nabla \log \widehat{p}_{\lambda_b(t)}(Y)\|^2 \\ &= -\frac{\lambda_b'(t) a(t)^2}{2} \mathbb{E}_{Z \sim q_t} \|\nabla \log q_t(Z) - \nabla \log p_t(Z)\|^2. \end{aligned} \tag{44}$$

The coefficient reduces to

$$\frac{\lambda_b'(t) a(t)^2}{2} = \frac{(1-b)^2 t}{(1-t)^3} \frac{(1-t)^2}{(1-b)^2} = \frac{t}{1-t}. \tag{45}$$

Finally substitute the velocity–score relation equation 29:

$$\begin{aligned} K'(t) &= -\frac{t}{1-t} \mathbb{E}_{Z \sim q_t} \|\nabla \log q_t(Z) - \nabla \log p_t(Z)\|^2 \\ &= -\frac{t}{1-t} \frac{(1-t)^2}{t^2} m(t) \\ &= -\frac{1-t}{t} m(t). \end{aligned} \tag{46}$$

□

Theorem A.2 (Decomposition, bound, and self-consistency)**.** *Under Assumptions A1 through A4, at the fixed sampled $(\mathcal{T}, b)$ with $t_i \leq 1$, the exact decomposition is*

$$\boxed{D_{\mathrm{KL}}(q_b \| p_b) = D_b + \mathbb{E}\big[w_b(T)\|v_q(Z_T, T) - v_p(Z_T, T)\|^2\big]}. \tag{47}$$

For every $\eta > 0$, it yields

$$\boxed{D_{\mathrm{KL}}(q_b \| p_b) \leq D_b + (1+\eta)O_b + (1+\eta^{-1})R_b}. \tag{48}$$

In particular, setting $\eta = 1$ gives

$$\boxed{D_{\mathrm{KL}}(q_b \| p_b) \leq D_b + 2O_b + 2R_b}. \tag{49}$$

If $R_b = 0$, the decomposition becomes

$$\boxed{D_{\mathrm{KL}}(q_b \| p_b) = D_b + O_b}. \tag{50}$$

Proof. Fix $b < s < \tau < t_i$. Integrating Proposition A.1 on $[s, \tau]$ gives

$$K(s) - K(\tau) = \int_s^\tau \frac{1-t}{t} m(t)\,\mathrm{d}t. \tag{51}$$

The integrand is nonnegative. Let $s \downarrow b$. Assumption A2 gives the limit of $K(s)$, and monotone convergence gives the limit of the integral:

$$D_{\mathrm{KL}}(q_b \| p_b) = K(\tau) + \int_b^\tau \frac{1-t}{t} m(t)\,\mathrm{d}t. \tag{52}$$

Both terms on the right are nonnegative, and their sum is finite. Now average over $\tau = T \sim \mathrm{Unif}[b, t_i]$ and apply equation 32 with $h(t) = (1-t)m(t)/t$:

$$\begin{aligned} D_{\mathrm{KL}}(q_b \| p_b) &= \mathbb{E}_T K(T) + \mathbb{E}_T\left[\int_b^T \frac{1-t}{t} m(t)\,\mathrm{d}t\right] \\ &= D_b + \mathbb{E}_T\left[(t_i - T)\frac{1-T}{T} m(T)\right] \\ &= D_b + \mathbb{E}_T\big[w_b(T)\mathbb{E}_{Z_T|T}\|v_q(Z_T, T) - v_p(Z_T, T)\|^2\big] \\ &= D_b + \mathbb{E}\big[w_b(T)\|v_q(Z_T, T) - v_p(Z_T, T)\|^2\big]. \end{aligned} \tag{53}$$

The first equality averages equation 52. The second uses the uniform-endpoint identity and the definition of D_b. The third substitutes the definitions of w_b and m, with the conditional law from equation 16. The last equality is the law of total expectation. This proves equation 47.

For the bound, omit the shared arguments (Z_T, T). By Assumption A3, $v_p = v_\phi$. Write $A = v_\theta - v_\phi$ and $B = v_q - v_\theta$. For $\eta > 0$, use $2\langle A, B\rangle \leq \eta\|A\|^2 + \eta^{-1}\|B\|^2$ to obtain

$$\begin{aligned} \|v_q - v_p\|^2 &= \|v_q - v_\phi\|^2 = \|A + B\|^2 \\ &= \|A\|^2 + 2\langle A, B\rangle + \|B\|^2 \\ &\leq (1+\eta)\|A\|^2 + (1+\eta^{-1})\|B\|^2 \\ &= (1+\eta)\|v_\theta - v_\phi\|^2 + (1+\eta^{-1})\|v_\theta - v_q\|^2. \end{aligned} \tag{54}$$

Multiply by $w_b(T) \geq 0$, take expectations, and substitute equation 47:

$$\begin{aligned} D_{\mathrm{KL}}(q_b \| p_b) - D_b &= \mathbb{E}\big[w_b(T)\|v_q - v_p\|^2\big] \\ &\leq (1+\eta)\mathbb{E}\big[w_b(T)\|v_\theta - v_\phi\|^2\big] \\ &\quad + (1+\eta^{-1})\mathbb{E}\big[w_b(T)\|v_\theta - v_q\|^2\big] \\ &= (1+\eta)O_b + (1+\eta^{-1})R_b. \end{aligned} \tag{55}$$

This proves equation 48; setting $\eta = 1$ gives equation 49. If $R_b = 0$, its nonnegative integrand is zero almost surely:

$$w_b(T)\|v_\theta - v_q\|^2 = 0 \qquad \text{a.s.} \tag{56}$$

Where $w_b(T) > 0$, this implies $v_\theta = v_q$. Where the weight is zero, the weighted discrepancies below vanish. Using teacher consistency again gives

$$\begin{aligned} \mathbb{E}\big[w_b(T)\|v_q - v_p\|^2\big] &= \mathbb{E}\big[w_b(T)\|v_\theta - v_p\|^2\big] \\ &= \mathbb{E}\big[w_b(T)\|v_\theta - v_\phi\|^2\big] = O_b. \end{aligned} \tag{57}$$

Substituting this equality into equation 47 proves equation 50. □

A.4 Training Objectives and Low-Noise Truncation

Conditional on the sampled grid and base, the TDM distribution-matching objective corresponds directly to D_b. OPD velocity regression on the same Z_T, with the weight in equation 17 and the teacher in Assumption A3, corresponds directly to O_b. The self-consistency residual R_b measures the discrepancy between the student network velocity and the velocity induced by its own forward-noised distribution. Both objectives share the student parameters, so D_b, O_b, and R_b can change together during training. Theorem A.2 relates these quantities to the base KL, and identifies their exact sum when $R_b = 0$.

Why low-noise velocity MSE. For a fixed interval endpoint t_i, the weight satisfies

$$w_b(t) = \frac{t_i}{t} - t_i - 1 + t, \tag{58}$$

$$w_b'(t) = 1 - \frac{t_i}{t^2} < 0, \qquad b < t < t_i \le 1, \tag{59}$$

$$w_b(t_i) = 0, \qquad w_b(t) \sim \frac{t_i}{t} \quad (b = 0,\ t \downarrow 0). \tag{60}$$

The strict inequality follows from $t^2 < t_i^2 \le t_i$. Thus the weight decreases toward the noisier end of each interval, while the factor $(1-t)/t$ also emphasizes small absolute noise levels. In equation 47, D_b is the KL at a random smoothing level, and the remaining dissipation is the weighted discrepancy between v_q and v_p. Under teacher consistency, equation 48 bounds this remainder by $(1+\eta)O_b + (1+\eta^{-1})R_b$; when $R_b = 0$, it equals O_b. Weighted velocity MSE therefore supplies an explicit objective for this remainder under self-consistency, with larger weights at lower noise levels. This motivates low-noise velocity supervision. The fixed threshold 0.5 is a design choice rather than a consequence of the theorem. Keeping only times below that threshold approximates the full O_b by its first term in

$$\begin{aligned} O_b = \mathbb{E}\big[w_b(T)\|v_\theta - v_\phi\|^2 \mathbf{1}_{\{T \le 0.5\}}\big] \\ + \mathbb{E}\big[w_b(T)\|v_\theta - v_\phi\|^2 \mathbf{1}_{\{T > 0.5\}}\big]. \end{aligned} \tag{61}$$

where all velocities are evaluated at (Z_T, T). The second term is the omitted contribution and remains part of the complete bound.

Numerical truncation. If numerical truncation removes the interval (b, δ) for some $b < \delta < t_i$, then equation 52 gives

$$D_{\mathrm{KL}}(q_b \| p_b) = D_{\mathrm{KL}}(q_\delta \| p_\delta) + \int_b^\delta \frac{1-t}{t} m(t)\, \mathrm{d}t. \tag{62}$$

The integral is the low-noise dissipation residual omitted by truncation. A decomposition based on the retained noise interval must include this residual and use the sampling normalization for that interval. In particular, truncating levels near zero does not remove this term from the exact identity.

B Implementation details for the reported static-routing variant

The reported four-step result uses static source routing and low-noise velocity MSE. Its SFT-augmented additive initialization is a rank-160, alpha-64 LoRA over 23 target modules. The initialization is merged into the base before fresh rank-32, alpha-64 all-linear student and fake adapters are created. We train with FSDP on four GPUs, a per-rank batch size of 64, and global batch size 256. The student and fake networks use learning rates of 2×10^{-6} and 1×10^{-5}, respectively, AdamW with $(\beta_1, \beta_2) = (0, 0.999)$ and weight decay 10^{-2}, and 200-step optimizer warmups. The loader cycles through the three training sources with equal frequency, yielding a homogeneous-source mini-batch at every step. The frozen initial student supplies the self-distillation target during the first 200 steps; afterward, source routing sends the whole mini-batch to one of the three frozen specialists. Rollouts use the randomized four-step schedule and interpolated re-noising that Sec. 4.1 describes.

The distribution-matching loss $\mathcal{L}_{\mathrm{TDM}}(\tilde{v}^{\mathrm{tgt}})$ that Alg. 1 accumulates is a differentiable surrogate for the gradient estimator of Eq. 4. Instead of backpropagating through the score difference $s_\psi - s_{\phi,\tau}$ directly, the algorithm first forms the velocity difference at the supervision state, then projects it back

Algorithm 1: Main-configuration training procedure for `StepX-OPD`.

Input: learning rate λ, training sources $\{\mathcal{C}_d\}$ with routing r, grid size N, velocity-loss weight w_{v}, teacher-warmup steps s_{w}, velocity-loss start step s_{v} and batch-mean threshold τ_{v}, student v_θ^{stu}, fake velocity v_ψ^{fake}, frozen initial student $v_{\theta_0}^{\mathrm{stu}}$, frozen specialists $\{v_m^{\mathrm{tea}}\}_{m=1}^{M}$

Output: optimized student v_θ^{stu}

for $s \leftarrow 0$ **to** $S-1$ **do**
- Sample source d and mini-batch $\mathcal{B}_d = \{c_j\}_{j=1}^{B}$ from $\mathcal{C}_d$, $\mathbf{c} \leftarrow (c_j)_{j=1}^{B}$;
- **if** $s < s_{\mathrm{w}}$ **then**
  - $v^{\mathrm{tgt}} \leftarrow v_{\theta_0}^{\mathrm{stu}}$;
- **else**
  - $v^{\mathrm{tgt}} \leftarrow v_{r(d)}^{\mathrm{tea}}$;
- Sample random grid $\boldsymbol{\sigma} \sim \mathcal{G}_N$, where $1 = \sigma_0 > \cdots > \sigma_N = 0$; `// random grid, Sec. 4.1`
- Sample noise $\bar{z}_{\sigma_0}$ from standard normal distribution;
- $\{\bar{z}_{\sigma_n}\}_{n=0}^{N} \leftarrow \mathrm{Rollout}^{\mathrm{sg}}(v_\theta^{\mathrm{stu}}, \mathbf{c}; \boldsymbol{\sigma}, \bar{z}_{\sigma_0})$;
- Sample z_{start} from $\{\bar{z}_{\sigma_n}\}_{n=0}^{N-1}$, with corresponding interval (σ_u, σ_ℓ);
- $z_{\mathrm{end}} \leftarrow z_{\mathrm{start}} + (\sigma_\ell - \sigma_u) v_\theta^{\mathrm{stu}}(z_{\mathrm{start}}, \sigma_u, \mathbf{c})$; `// Eq. 2`
- Sample base timestep τ_{base} from σ_ℓ to σ_u;
- $x_{\mathrm{base}} \leftarrow \mathrm{Interpolate}(z_{\mathrm{start}}, \mathrm{sg}[z_{\mathrm{end}}], \tau_{\mathrm{base}})$; `// Eq. 6`
- Sample timestep τ from τ_{base} to σ_u;
- $z_\tau \leftarrow \mathrm{ReNoise}(x_{\mathrm{base}}, \tau)$; `// Eq. 7`
- $\psi \leftarrow \mathrm{UpdateFake}(\psi; z_\tau, \tau, \mathbf{c})$; `// default TDM fake update`
- $\tilde{v}^{\mathrm{tgt}} \leftarrow v_{\mathrm{cfg}}^{\mathrm{tgt}}(z_\tau, \tau, \mathbf{c})$;
- $\Delta_\tau \leftarrow \tilde{v}^{\mathrm{tgt}} - v_\psi^{\mathrm{fake}}(z_\tau, \tau, \mathbf{c})$;
- $\hat{z} \leftarrow \mathrm{sg}[z_{\mathrm{end}} - \tau \Delta_\tau]$, $\quad \mathcal{L}_{\mathrm{TDM}} \leftarrow \rho(z_{\mathrm{end}} - \hat{z})$;
- $\mathcal{L}_{\mathrm{v}} \leftarrow 0$;
- $\bar{\tau} \leftarrow B^{-1} \sum_{j=1}^{B} \tau_j$;
- **if** $s \geq s_{\mathrm{v}}$ ***and*** $\bar{\tau} \leq \tau_{\mathrm{v}}$ **then**
  - $\mathcal{L}_{\mathrm{v}} \leftarrow \mathrm{mean}\Big[\big(v_\theta^{\mathrm{stu}}(z_\tau, \tau, \mathbf{c}) - \mathrm{sg}[\tilde{v}^{\mathrm{tgt}}]\big)^{\odot 2}\Big]$; `// Eq. 9`
- $\theta \leftarrow \theta - \lambda \nabla_\theta \left(\mathcal{L}_{\mathrm{TDM}} + w_{\mathrm{v}} \mathcal{L}_{\mathrm{v}}\right)$;

onto the differentiable endpoint state z_{end} as a stop-gradient pseudo-target, and finally regresses z_{end} toward that pseudo-target with a pseudo-Huber loss:

$$\Delta_\tau = \tilde{v}^{\mathrm{tgt}} - v_\psi^{\mathrm{fake}}(z_\tau, \tau, c), \qquad \hat{z} = \mathrm{sg}[z_{\mathrm{end}} - \tau \Delta_\tau], \qquad \mathcal{L}_{\mathrm{TDM}}(\tilde{v}^{\mathrm{tgt}}) = \rho(z_{\mathrm{end}} - \hat{z}), \tag{63}$$

where, for packed tokens $p \in P$, $\rho(r) = |P|^{-1} \sum_{p \in P} (\sqrt{\|r_p\|_2^2 + c_p^2} - c_p)$ and c_p is the implementation's token-scale parameter. Because $\hat{z}$ is detached, this residual carries the target/fake velocity difference into the student update through z_{end}.

For completeness, the compact operators in Alg. 1 denote the following implementation. In particular, $\mathrm{UpdateFake}$ updates the fake velocity v_ψ^{fake} once at the currently constructed state (z_τ, τ) using the TDM-style fake-score objective, terminal-aware $\mathrm{CleanTarget}$, and stabilized conditional importance ratio defined below. Given the detached endpoint $\bar{z} = \mathrm{sg}[z_{\mathrm{end},j}]$, the terminal-aware clean target is

$$\mathrm{CleanTarget}_\theta(\bar{z}, \sigma_\ell, c_j) = \begin{cases} \bar{z}, & \sigma_\ell = 0, \\ \mathrm{sg}[\bar{z} - \sigma_\ell v_\theta^{\mathrm{stu}}(\bar{z}, \sigma_\ell, c_j)], & \sigma_\ell > 0. \end{cases} \tag{64}$$

The interpolation and re-noising lines of Alg. 1 apply Eqs. (6)–(7) and yield the detached tuple $(x_{\mathrm{base},j}, \tau_{\mathrm{base},j}, z_{\tau_j}, \tau_j)$. For the stabilized conditional importance ratio, define

$$\begin{aligned} \tau_j^c &= \mathrm{clip}(\tau_j, 10^{-5}, 1), & b_j^c &= \mathrm{clip}(\tau_{\mathrm{base},j}, 0, 1 - 10^{-5}), \\ \kappa_j^c &= \frac{1 - \tau_j^c}{1 - b_j^c}, & (\beta_j^c)^2 &= \max\big((\tau_j^c)^2 - (\kappa_j^c b_j^c)^2, 10^{-10}\big), \\ q_j^+ &= \mathcal{N}\big((1 - \tau_j^c) x_{0,j}^{\mathrm{tgt}}, (\tau_j^c)^2 I\big), & q_j^- &= \mathcal{N}\big(\kappa_j^c x_{\mathrm{base},j}, (\beta_j^c)^2 I\big), \\ \ell_j &= \big\langle \log q_j^+(z_{\tau_j}) - \log q_j^-(z_{\tau_j}) \big\rangle_{\mathrm{lat}}, & \mathrm{StabCondIS}_j &= \exp(\mathrm{clip}(\ell_j, -20, 20)). \end{aligned} \tag{65}$$

Here $\langle \cdot \rangle_{\mathrm{lat}}$ averages over each packed sample's latent elements, exactly matching the stabilized ratio used in training.

Three frozen DiffusionOPD teachers provide the GenEval, OCR, and aesthetic velocity fields. A static lookup routes the GenEval, OCR, and PickScore training sources to these three teachers, respectively. After warmup, the routed teacher supplies both the distribution-matching target and its own guided velocity for the auxiliary MSE. The velocity-MSE term has weight 0.5, starts at step 500, and, when the per-rank batch mean sampled time satisfies $\bar{\tau} \leq 0.5$, applies to the entire batch. The TDM residual uses the configured pseudo-Huber reduction. The endpoint objective uses the active target: the frozen initial student during the first 200 steps and the routed specialist afterward. The run has no separate quality teacher. The reported result and the accompanying analysis therefore cover static source routing rather than reward-based or student-competitive routing.

Sec. 5.4 reports comparisons of velocity-regression coverage, its starting step, and the SFT merge. Tab. 6 compares the stated initialization-merge variants, while Tab. 5 compares the stated coverage and starting-step variants. The implementation also supports training on 8 NVIDIA H800 GPUs, with wall-clock time determined by the selected hardware configuration.

Every evaluated model is sampled at 512×512 resolution using one of two fixed settings: full-step models run forty steps at guidance scale 4.5, and four-step students, including `StepX-OPD`, run four steps at guidance scale 1.0. Fig. 1 and Fig. 4 draw on these same two settings, and the two prompt sets written for the recurrence measurement follow them as well, with Tab. 14 sampling both sets at the four-step, guidance-scale-1.0 setting across the six four-step models it evaluates. These two settings state guidance as a classifier-free guidance scale, which is what SD3.5-Medium uses. FLUX.1-dev instead carries a distilled guidance embedding, so its forty-step rows set that embedding to 3.5 and its four-step rows set it to 1.0, with classifier-free guidance disabled throughout.

C Evaluation protocol

For every evaluated model, we generate four images for each GenEval record, OCR prompt, and PickScore prompt. See App. B for the sampling steps, guidance scale, and resolution used to generate them. GenEval's 2,212 records include repeated prompt strings, so they correspond to 553 unique strings rather than 2,212 independent prompts. The resulting image counts are 8,848 for GenEval, 4,072 for OCR, and 8,192 for PickScore. DeQA and the five additional measures that Tab. 7 reports are evaluated on the same fixed 250-image subset from each of GenEval, OCR, and PickScore, for 750 images total. GenEval uses the official detector-based scorer; OCR uses normalized target-string matching with PaddleOCR; and PickScore is reported both in its raw scale and, when forming Average, after division by 26. CLIPScore is reported after division by 30. Average is $(\text{GenEval} + \text{OCR} + \text{DeQA}/5 + \text{PickScore}/26)/4$.

Sample sizes. The prompt-word recurrence measurement on real benchmark prompts draws on the 2,048 prompts of the PickScore benchmark, with three generations per prompt from three random seeds. We partition these prompts by whether a scene that faithfully renders the prompt would be expected to contain text, giving 1,633 prompts with no text expected (4,899 images per model), 128 that name an object which itself bears text (384 images), 193 that depict a scene where text is typical (579 images), and 94 that specify particular text content (282 images). Tab. 12 reports the first of these groups and Tab. 13 reports all four, both with PickScore after division by 26. Within the first of these, the group used in Fig. 4 and in Figs. 5, 7 and 8 is the further cut that also excludes prompts that name an existing entity or that describe an object able to carry text, which leaves 789 prompts and 2,367 images per model. Both panels of Fig. 4 draw on that cut, and its right panel sets the cut against the fourth group above, the 94 prompts that specify particular text content. Two further prompt sets, written for this measurement rather than drawn from the benchmark, appear in Tab. 14: a slogan register of 100 prompts phrased like a slogan or title, and a descriptive set of 100 prompts with nothing to write on. Tab. 14 draws four seeds per prompt from each set across the six four-step models it evaluates, giving $N = 400$ in both columns.

D Additional related work on unintended text rendering

Text rendered without being asked for. Work that touches this behavior reaches it from several directions. On the representation side, Materzyńska et al. (2022) show that CLIP's image encoder holds spelling and scene content in one space, and that text-to-image methods guided by CLIP spell

Table 7: Comparison of additional aligned quality and preference measures across methods, for both base architectures evaluated.

Base Model	Model	NFEs	CLIPScore ↑	HPSv2.1 ↑	Aesthetic ↑	ImageReward ↑	UnifiedReward ↑
SD3.5-Medium	Base	80	1.000	0.289	5.552	1.081	3.701
	GenEval Teacher	80	1.020	0.277	5.541	1.222	3.783
	OCR Teacher	80	0.994	0.274	5.390	1.015	3.579
	PickScore Teacher	80	1.004	0.325	5.954	1.396	3.964
	DeQA Teacher	80	1.011	0.301	5.622	1.293	3.903
	DiffusionOPD	80	1.033	0.304	5.676	1.288	3.874
	Flow-OPD	80	1.021	0.313	5.788	1.389	3.992
	Base + TDM	4	0.983	0.285	5.567	0.987	3.524
	Flow-OPD + TDM	4	1.012	0.304	5.811	1.344	3.880
	DiffusionOPD + TDM	4	1.026	0.307	5.893	1.263	3.849
	`StepX-OPD`	4	1.024	0.307	5.772	1.301	3.894
FLUX.1-dev	Base	40	0.969	0.299	5.784	1.086	3.769
	GenEval Specialist	40	0.997	0.303	5.795	1.292	3.902
	OCR Specialist	40	0.982	0.298	5.713	1.138	3.798
	PickScore Specialist	40	1.000	0.325	6.101	1.412	3.942
	Joint Flow-OPD	40	1.017	0.310	5.790	1.355	3.987
	Joint DiffusionOPD	40	1.013	0.310	5.804	1.332	3.981
	Base + TDM	4	0.986	0.297	5.679	1.107	3.515
	Flow-OPD + TDM	4	1.017	0.308	5.642	1.327	3.800
	DiffusionOPD + TDM	4	1.020	0.307	5.662	1.329	3.771
	`StepX-OPD`	4	1.021	0.309	5.794	1.344	3.923

out the single word they are conditioned on, in a setting where generation is an iterative optimization through CLIP rather than a trained sampler. SpectralCLIP (Xu et al., 2024) names these written words textual artifacts and suppresses them by masking the frequency band of the CLIP embedding they occupy, a remedy that acts on the CLIP guidance path a flow matching sampler does not pass through. Censored sampling (Yoon et al., 2023) treats the behavior as something to remove at sampling time: Stable Diffusion, asked for a photo of a human, sometimes produces prominent embedded text or shows text in place of any human figure, and about 23.7% of the samples drawn from that one hand-written prompt are judged malign, which ties the measurement to the prompt rather than to a benchmark a model is scored on. TangPoetryBench (Hu et al., 2026) instead scores it under prompts that instruct the model to include no text, giving text integrity a dimension of its own that separates leakage of the prompt's own text from garbled characters and counting 119 of 1,280 images as carrying a text issue, so what it measures there is disobedience of an explicit instruction. On the reinforcement learning side, DDPO (Black et al., 2024) reports a policy trained against a vision-language reward printing the string `sixx ttutttas` over an image of eight turtles, presented as one qualitative case from a single-reward run, and RLRF (Rodriguez et al., 2025) lists a text-in-image hack in which the policy emits the prompt string as an SVG text primitive rather than drawing the described scene. Hong et al. (2026) organize the failure modes of text-to-image reinforcement learning around artifact-prone generation and train an artifact reward model as a regularizer for the reward models in use, and rendered text is not among the modes that catalogue lists. Flow-OPD (Fang et al., 2026) attributes degraded aesthetics under aggressive optimization of text rendering to reward hacking, citing Flow-GRPO (Liu et al., 2025), and the evidence it reports for the effect is a drop in GenEval, which reads object composition rather than what the image writes. We report this behavior as a count taken under one protocol on benchmark prompts that neither ask for rendered text nor forbid it, on the same backbone before and after several rewards are integrated into it.

E Supplementary results

Additional quality measures. Tab. 7 reports, for the models of both base architectures, aligned quality and preference measures that lie outside the Average score.

Teacher–base velocity displacement. Tab. 8 summarizes a shared-reference diagnostic for six task specialists. At each noise level, we compute

$$D_m(t) = \frac{N^{-1}\sum_{i=1}^{N}\|v_m(x_{t,i}) - v_b(x_{t,i})\|_2}{N^{-1}\sum_{i=1}^{N}\|v_b(x_{t,i})\|_2}.$$

Table 8: Measurement of mean normalized teacher–base velocity displacement, $\mathrm{mean}_t\, D_m(t)$, for six task specialists.

Family	Specialist	$\mathrm{mean}_t\, D_m(t)$
Flow-GRPO	GenEval	0.1462
Flow-GRPO	OCR	0.1529
Flow-GRPO	PickScore	0.1322
DiffusionOPD	GenEval	0.5051
DiffusionOPD	OCR	0.1330
DiffusionOPD	Aesthetic	0.2623

Table 9: Measurement of the forty-step full-benchmark results for the SFT checkpoints used as pre-distillation initialization references.

Initialization reference	GenEval ↑	OCR ↑	DeQA ↑	PickScore ↑	Average ↑
GenEval-only SFT checkpoint	0.9153	0.6152	4.2299	21.79	0.8037
GPT-image SFT checkpoint	0.8977	0.5203	4.1782	22.31	0.7779

Here $N = 1{,}536$ prompts comprises 512 each from GenEval, OCR, and PickScore. Every specialist and the base model receive the same base-reference state, prompt, noise, and CFG 4.5 at each noise level. This diagnostic records velocity displacement; causal conclusions and student-quality measurements are outside its scope.

Pre-distillation SFT reference. Tab. 9 gives two forty-step SFT checkpoints as initialization references, one trained on GenEval data and one on GPT-image data. Both are separate from the four-step distillation comparisons.

Initialization recipe and teacher family. Tab. 10 reports four completed four-step full-benchmark evaluations crossing two initialization recipes with two multi-teacher on-policy distillation teacher families. The averaged-specialist merge pools several published single-reward LoRA adapters with uniform weight over the attention projections, while the SFT-augmented merge instead adds a supervised fine-tuning increment to a similarly pooled combination that also covers the feed-forward and embedding modules. Paired with Flow-GRPO teachers, the averaged-specialist merge reaches GenEval 0.885, OCR 0.874, DeQA 4.073, PickScore 22.61, and Average 0.8610, and paired with DiffusionOPD teachers it reaches 0.903, 0.896, 4.164, 22.89, and 0.8779 on the same columns. The SFT-augmented merge paired with Flow-GRPO teachers reaches 0.932, 0.886, 4.077, 22.75, and 0.8772, and paired with DiffusionOPD teachers it reaches 0.944, 0.899, 4.170, 22.98, and 0.8901.

Training-source sampling ratio. Tab. 11 ablates the sampling ratio over the three training sources, GenEval, OCR, and PickScore, within each of two student configurations. The two configurations differ in initialization and in whether the precise teacher and the low-noise velocity regression are enabled, so only the rows sharing a configuration are comparable; the two configurations themselves are not. Raising the OCR source's weight from 1:1:1 to 1:2:1 raises the OCR score in both configurations, from 0.5776 to 0.6830 for the fresh-LoRA student and from 0.7902 to 0.8154 for the merged-initialization student, and the Average score rises alongside it in both cases.

Prompt-word recurrence on prompts with no text called for. Tab. 12 reports what seven models write into the image on prompts that ask for no rendered text, six of them sampled at forty steps and `StepX-OPD` at its four-step working point. The prompts are the 2,048 that ship with the PickScore benchmark, stratified by a rule that reads the prompt text alone and was fixed before any model output was generated, and the stratum reported here is the one where no text is expected, whose described scenes hold no surface able to carry text. We read every generation with the same OCR service that supplies the training reward, and count a generation as prompt-word recurrence when its detected text contains an adjacent pair of content words from the prompt, or any six-character contiguous substring of the prompt. The untrained SD3.5-Medium base renders text in 166 generations and reproduces a prompt word in 21 of them, while SD3.5-Medium DiffusionOPD renders text in 699 and reproduces a prompt word in 218. Since the stratification reads the prompt

Table 10: Ablation of two initialization recipes crossed with two multi-teacher on-policy distillation teacher families, reported as full-benchmark scores.

Initialization	Teachers	GenEval ↑	OCR ↑	DeQA ↑	PickScore ↑	Average ↑
Averaged-specialist merge	Flow-OPD	0.885	0.874	4.073	22.61	0.8610
	DiffusionOPD	0.903	0.896	4.164	22.89	0.8779
SFT-augmented merge	Flow-OPD	0.932	0.886	4.077	22.75	0.8772
	DiffusionOPD	0.944	0.899	4.170	22.98	0.8901

Table 11: Ablation of the training-source sampling ratio (GenEval:OCR:PickScore) under two student configurations. Rows within a configuration share every other setting and are comparable; the two configurations differ in additional respects and are not comparable to each other.

Student configuration	Source ratio	GenEval ↑	OCR ↑	DeQA ↑	PickScore ↑	Average ↑
Fresh-LoRA student	1:1:1	0.752	0.5776	4.129	22.46	0.7548
	1:2:1	0.762	0.683	4.145	22.42	0.7839
Merged-init. student	1:1:1	**0.944**	**0.899**	**4.170**	**22.98**	**0.8901**
	1:2:1	0.869	0.815	4.095	22.53	0.8427

text alone, it sets aside the prompts that name a text-bearing object, and a model remains free to introduce a carrier the prompt never mentions: in four of the prompts counted here, the surface the model writes on appears nowhere in the prompt. Fig. 4 draws from a further cut of this stratum that also excludes prompts that name an existing entity or that describe an object able to carry text, Tab. 13 gives the same two counts on all four strata, and Fig. 6 resolves the measurement to individual prompts.

Recurrence against preference score. The last column places this measurement beside the preference score that these same prompts carry, on the divided scale that enters the Average score. On SD3.5-Medium the integrated model stands above its base on both columns: recurrence goes from 21 to 218 and PickScore goes from 0.838 to 0.912 alongside it. This score is coupled to the models it ranks: the teachers behind the two FLUX.1-dev integrated models reported in Tab. 12 are experts trained against a PickScore reward, the Flow-OPD cold-start weights contain that expert, and Flow-OPD selects its rollout samples by task reward. A reader who consults this score alone therefore has no view of the text these models render.

Prompt-word recurrence measurements. Tab. 13 extends Tab. 12 to all four prompt groups, so the counts on the group that calls for no rendered text can be read against the group whose prompts state what text should appear. Fig. 5 reports, for seven models evaluated under one protocol at forty sampling steps, the count of generated images whose detected text reproduces a word drawn from the prompt, on the benchmark prompt group Fig. 4 draws from. Tab. 14 reports that same measurement as a fraction for six four-step models, on a slogan register and a descriptive set with nothing to write on, both written for this measurement rather than drawn from the benchmark, together with their OCR and GenEval scores. Fig. 6 resolves the measurement to individual prompts for the three forty-step models of Fig. 4: each point places one prompt's recurrence rate over three seeds against that prompt's PickScore over the same three seeds, PickScore being the quality measure these same prompts carry.

Qualitative examples on the no-text-called-for subset. Figs. 7 and 8 extend the left panel of Fig. 4 in two directions. The rows grow from three models to seven, adding the four single-reward experts between the untrained base and the two published multi-capability models, so a reader can see which reward the rendered text follows. The columns are fourteen prompts of that subset on which DiffusionOPD renders a word from the prompt, and each prompt is printed in full below its block.

Table 12: Comparison of text rendered, the count of generations with any detected on-image text, and prompt-word recurrence, the count among those whose detected text reproduces a word from the prompt, across models, on PickScore benchmark prompts that do not call for rendered text, together with each model's PickScore.

Model	Text rendered	Prompt-word recurrence	PickScore ↑
SD3.5-Medium base	166	21	0.838
SD3.5-Medium SFT (GenEval)	283	40	0.838
SD3.5-Medium DiffusionOPD	699	218	0.912
SD3.5-Medium Flow-OPD	238	66	0.872
FLUX.1-dev base	177	8	0.844
FLUX.1-dev DiffusionOPD	266	44	0.907
FLUX.1-dev Flow-OPD	276	54	0.904
`StepX-OPD` (four-step)	304	94	0.885

Table 13: Comparison of text rendered, the count of generations with any detected on-image text, and prompt-word recurrence, the count among those whose detected text reproduces a word from the prompt, together with PickScore, across models, for four prompt groups defined by whether the described scene calls for rendered text.

	No text expected			Text-bearing object named			Scene-typical text			Specified text content		
Model	Text rendered	Prompt-word recurrence	Pick-Score ↑	Text rendered	Prompt-word recurrence	Pick-Score ↑	Text rendered	Prompt-word recurrence	Pick-Score ↑	Text rendered	Prompt-word recurrence	Pick-Score ↑
SD3.5-Medium base	166	21	0.838	149	54	0.821	74	2	0.842	234	126	0.860
SD3.5-Medium SFT (GenEval)	283	40	0.838	162	78	0.820	103	12	0.840	247	137	0.872
SD3.5-Medium DiffusionOPD	699	218	0.912	273	178	0.889	198	44	0.909	247	159	0.928
SD3.5-Medium Flow-OPD	238	66	0.872	159	77	0.850	80	11	0.870	255	155	0.895
FLUX.1-dev base	177	8	0.844	66	12	0.823	84	3	0.848	225	115	0.871
FLUX.1-dev DiffusionOPD	266	44	0.907	144	63	0.891	141	25	0.906	242	133	0.927
FLUX.1-dev Flow-OPD	276	54	0.904	151	60	0.887	156	17	0.902	242	132	0.928
`StepX-OPD` (four-step)	304	94	0.885	200	73	0.858	74	17	0.880	253	141	0.895

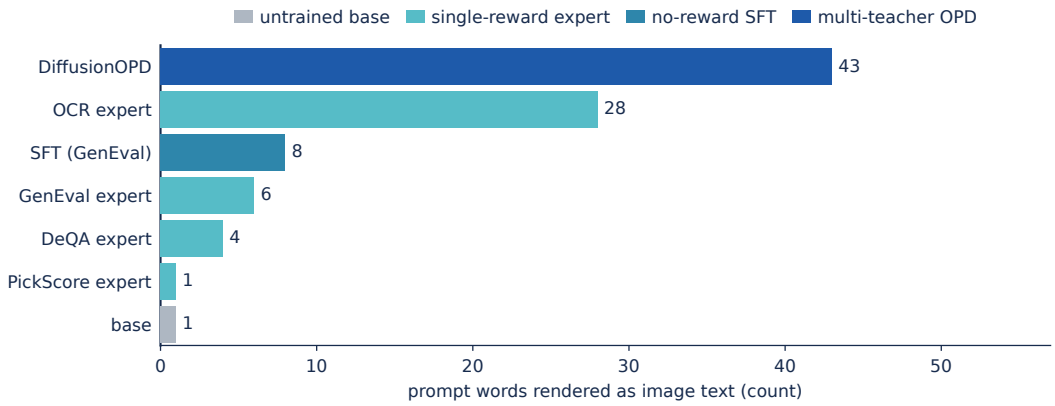


Figure 5: Count of generated images whose detected text reproduces a word drawn from the prompt, for seven models measured under one protocol at forty sampling steps, on the prompt group that calls for no rendered text ($N = 2{,}367$ images per model, the group Fig. 4 draws from). Colour marks the kind of post-training each model received.

Table 14: Comparison of prompt-word recurrence, the fraction of generations whose detected on-image text reproduces words from the prompt, across four-step models under two prompt registers, alongside each model's OCR and GenEval scores from the full evaluation.

	Prompt-word recurrence			
Model	Slogan register	Descriptive, nothing to write on	OCR ↑	GenEval ↑
Base + TDM	0.168	0.000	0.335	0.653
DiffusionOPD + TDM	0.782	0.000	0.833	0.838
Flow-OPD + TDM	0.675	0.000	0.751	0.875
SFT + TDM	0.610	0.000	0.725	0.932
`StepX-OPD`, w/o precise teacher	0.820	0.000	0.899	0.944
`StepX-OPD`, w/ precise teacher	0.610	0.000	0.895	0.945

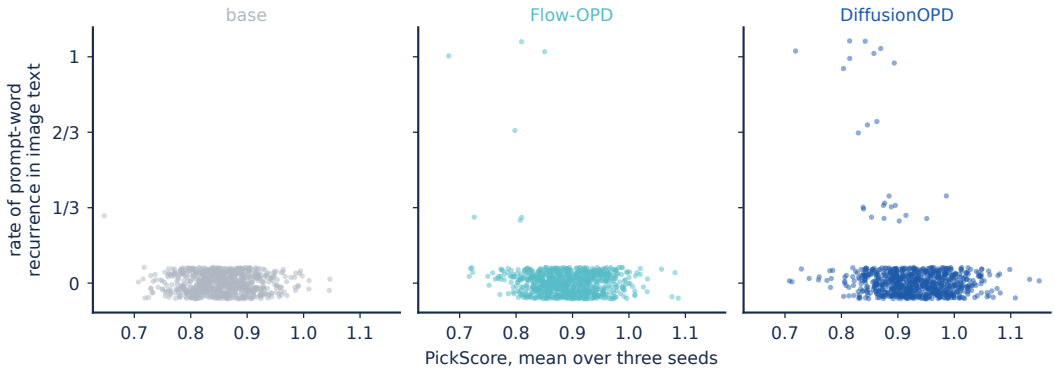
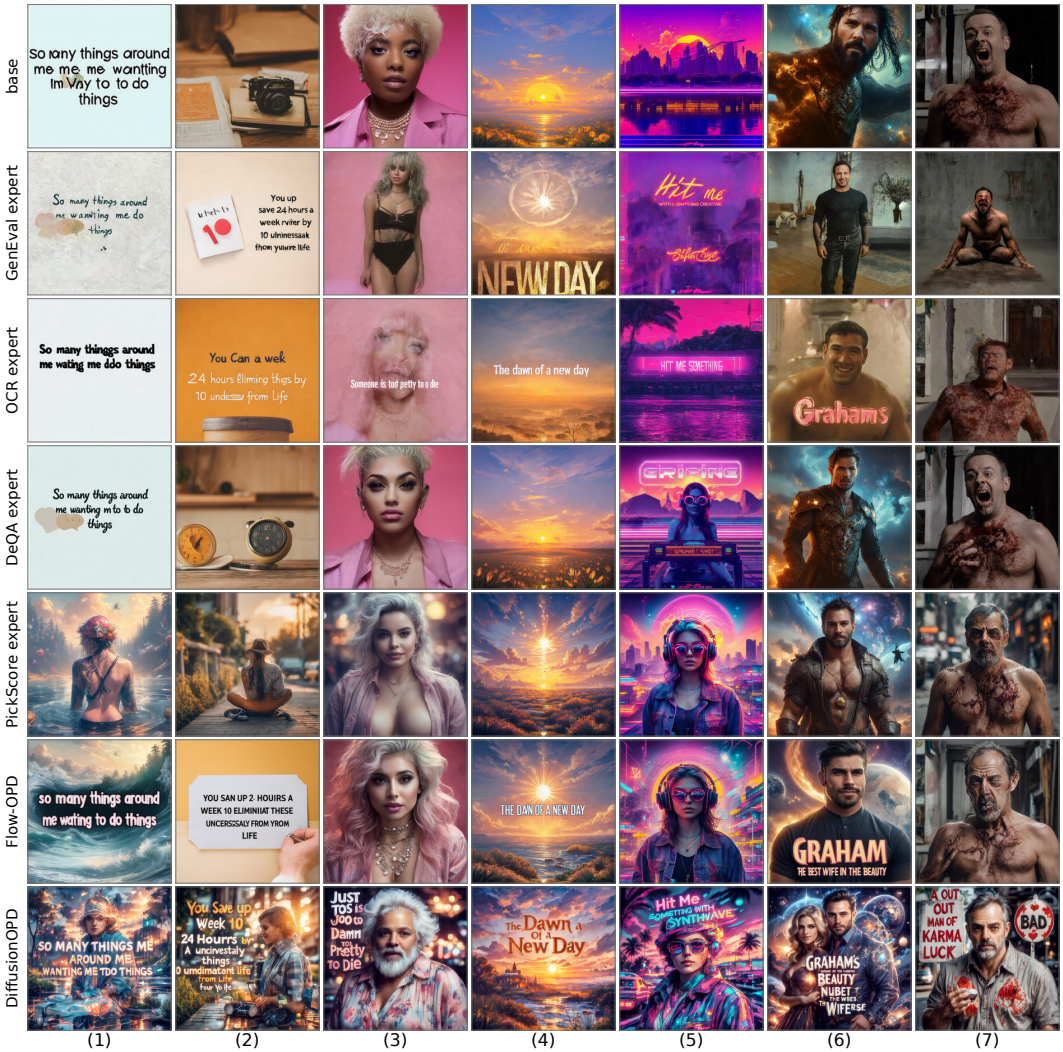


Figure 6: Prompt-word recurrence against PickScore, resolved to individual prompts. Each point is one prompt on one model: the vertical coordinate is the fraction of that prompt's three seeds whose detected on-image text reproduces a word from the prompt, and the horizontal coordinate is that prompt's PickScore averaged over the same three seeds. Panels and colours are the three SD3.5-Medium models of Fig. 4 at forty sampling steps. The prompt group is the one Fig. 4 draws from, 789 prompts at three seeds each ($N = 2{,}367$ images per model). The vertical coordinate takes only the values 0, 1/3, 2/3 and 1; points carry a small vertical jitter so that the points sharing a value are visible as separate marks.



(1) "so many things around me wanting me to do things 😅" (2) "You can save up to 24 hours a week by eliminating these 10 unnecessary things from your life." (3) "An image of someone who is just too damn pretty to die" (4) "the dawn of a new day" (5) "Hit me with something creative, in synthwave retro style" (6) "A man on a mission to summon the best wife in the universe, grahams number beauty" (7) "Photo of A man out of luck, bad karma, funny, humorous"

Figure 7: Text seven models render into the image, on seven of the prompts of the Fig. 4 subset for which DiffusionOPD produces an image whose text reproduces a word from the prompt. Columns are prompts (1)–(7), printed in full below the block; rows are the untrained base, four single-reward experts, and two published multi-capability models. Over the subset ($N = 2{,}367$ images per model) the count of images whose text reproduces a word from the prompt is 1 for the base, 6 for the GenEval expert, 28 for the OCR expert, 4 for the DeQA expert, 1 for the PickScore expert, 14 for Flow-OPD, and 43 for DiffusionOPD. Fig. 8 continues with prompts (8)–(14). All models are SD3.5-Medium at forty sampling steps.

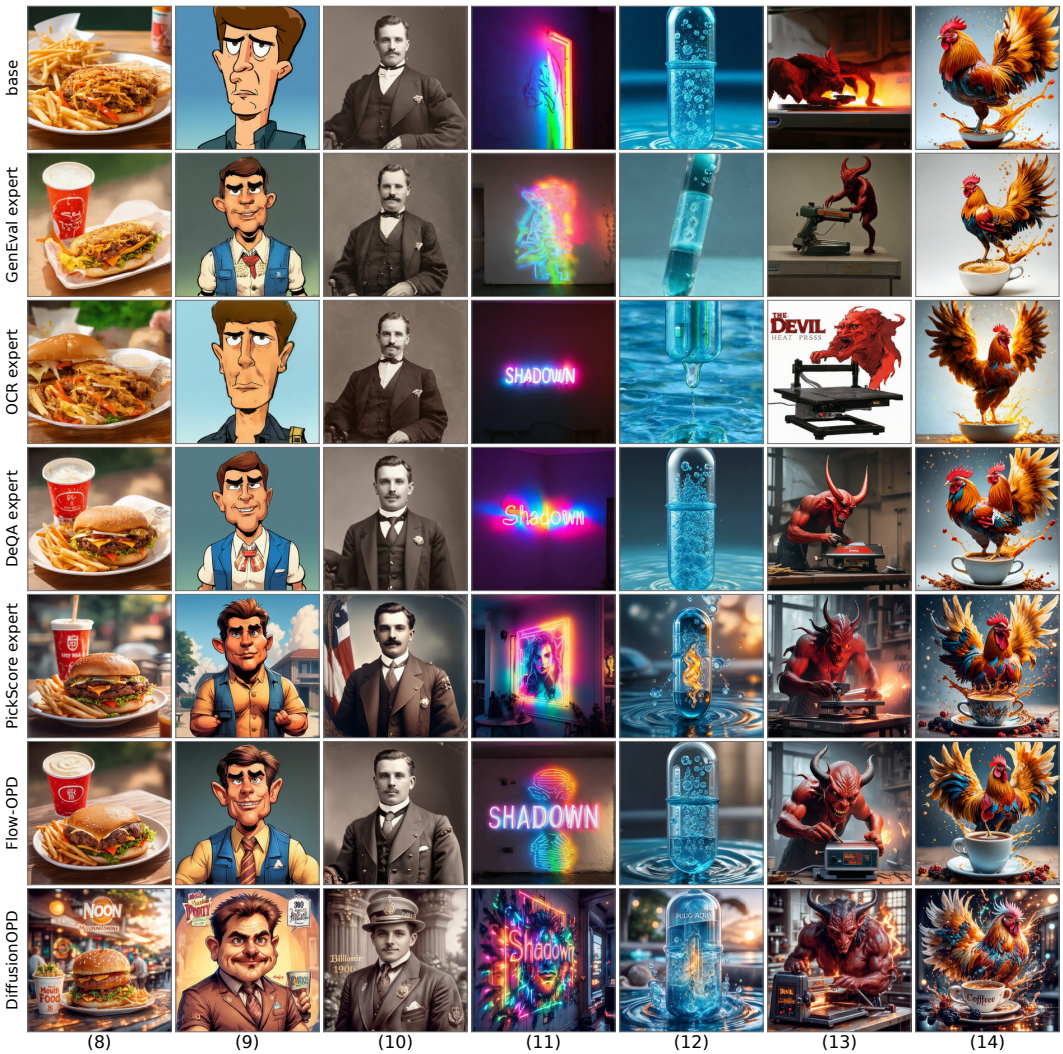


Figure 8: Text seven models render into the image, continuing Fig. 7 with prompts (8)–(14) of the same fourteen. Columns are those prompts, printed in full below the block; rows are the untrained base, four single-reward experts, and two published multi-capability models. Over the subset ($N = 2{,}367$ images per model) the count of images whose text reproduces a word from the prompt is 1 for the base, 6 for the GenEval expert, 28 for the OCR expert, 4 for the DeQA expert, 1 for the PickScore expert, 14 for Flow-OPD, and 43 for DiffusionOPD. All models are SD3.5-Medium at forty sampling steps.

F ADDITIONAL QUALITATIVE COMPARISONS

Fig. 9 through Fig. 12 extend the comparison in Fig. 3 with sixteen further self-written prompts, four per figure, on the same four-step models, rather than the six-model comparison on benchmark prompts that Fig. 2 gives. Fig. 9 and Fig. 10 show eight scenes that call for no rendered text. Fig. 11 keeps that register for two of its four prompts and turns to a short phrase rendered on an object for the other two, and Fig. 12 specifies a phrase to render in every one of its four prompts. Fig. 11(a) is a failure case: StepX-OPD renders the requested bottom-to-top bear–eagle–fish order as bear–fish–eagle, showing a spatial and hierarchical binding error.

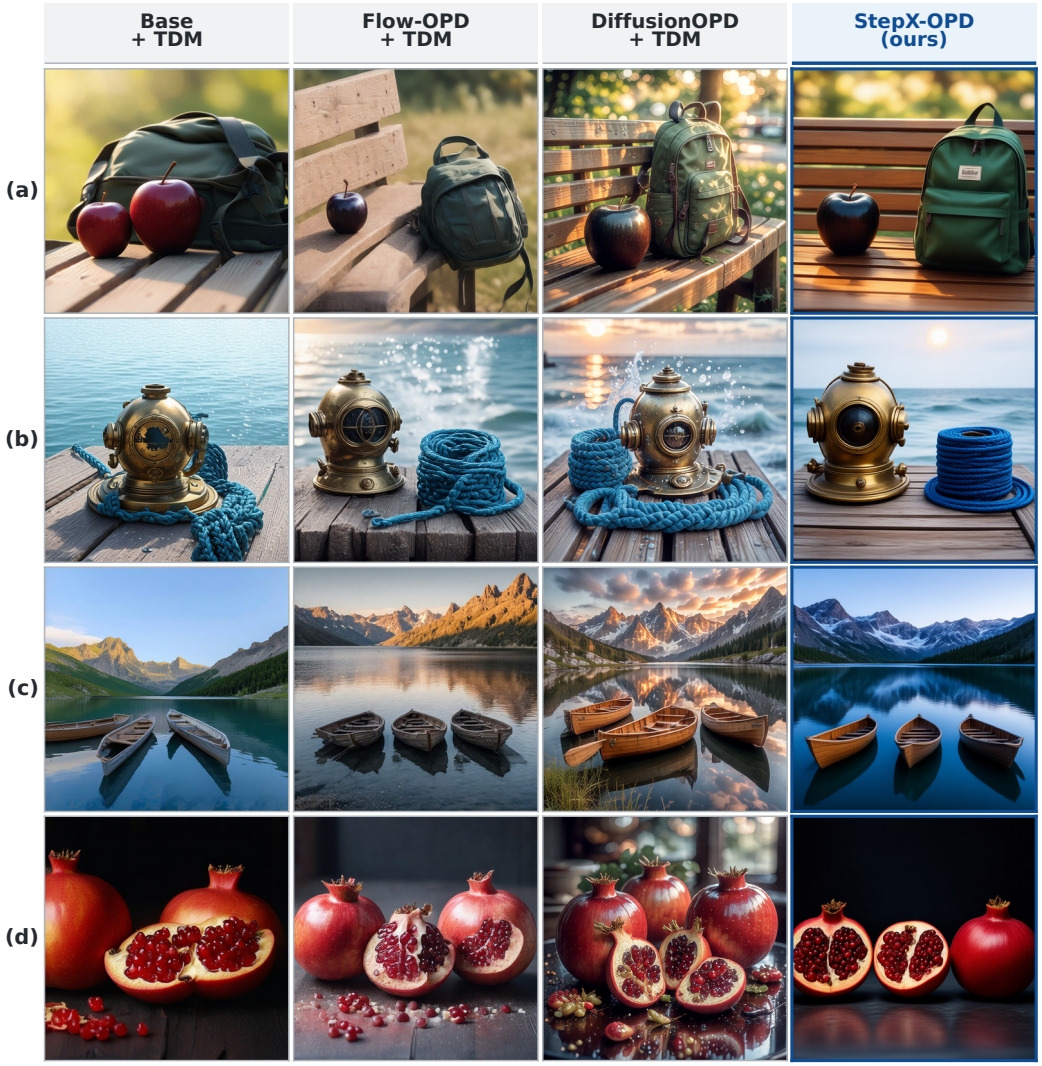


Figure 9: (a) A black apple and a green backpack on a sunlit wooden bench, shallow depth of field, warm afternoon light; (b) A brass diving helmet and a coil of blue rope on a weathered dock, sea spray, low sun behind; (c) Three wooden rowing boats moored on a still alpine lake at dawn, mountains mirrored in the water; (d) Three ripe pomegranates split open on a dark table, seeds glistening, dramatic side light, baroque still life.

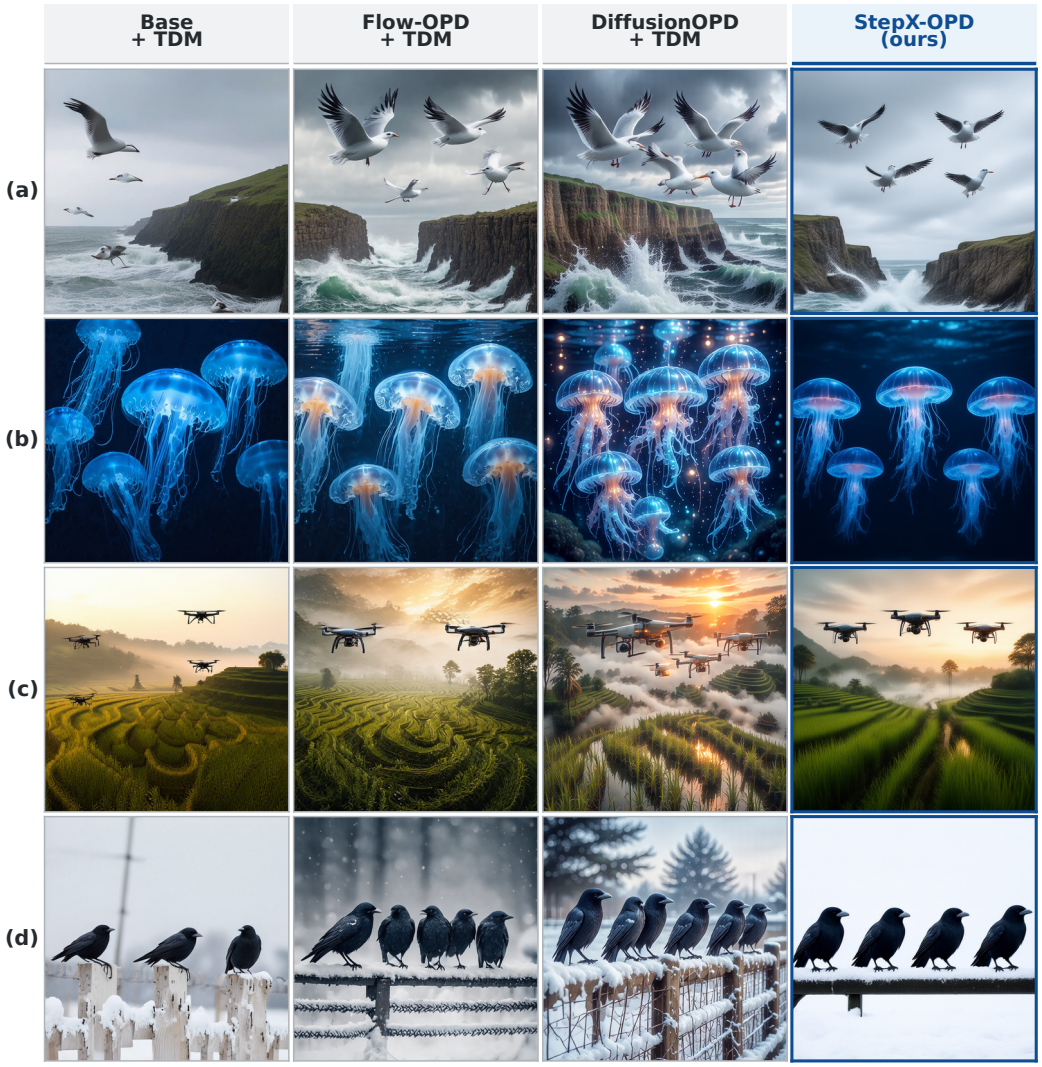


Figure 10: (a) Four seagulls in flight against a stormy grey sky above cliffs, wind-torn spray below; (b) Five jellyfish drifting in dark water, translucent bells glowing blue, fine tentacles trailing; (c) Three drones hovering above a terraced rice field at sunrise, mist between the terraces; (d) Four crows lined up on a snowy fence rail, black against white, flat overcast light.

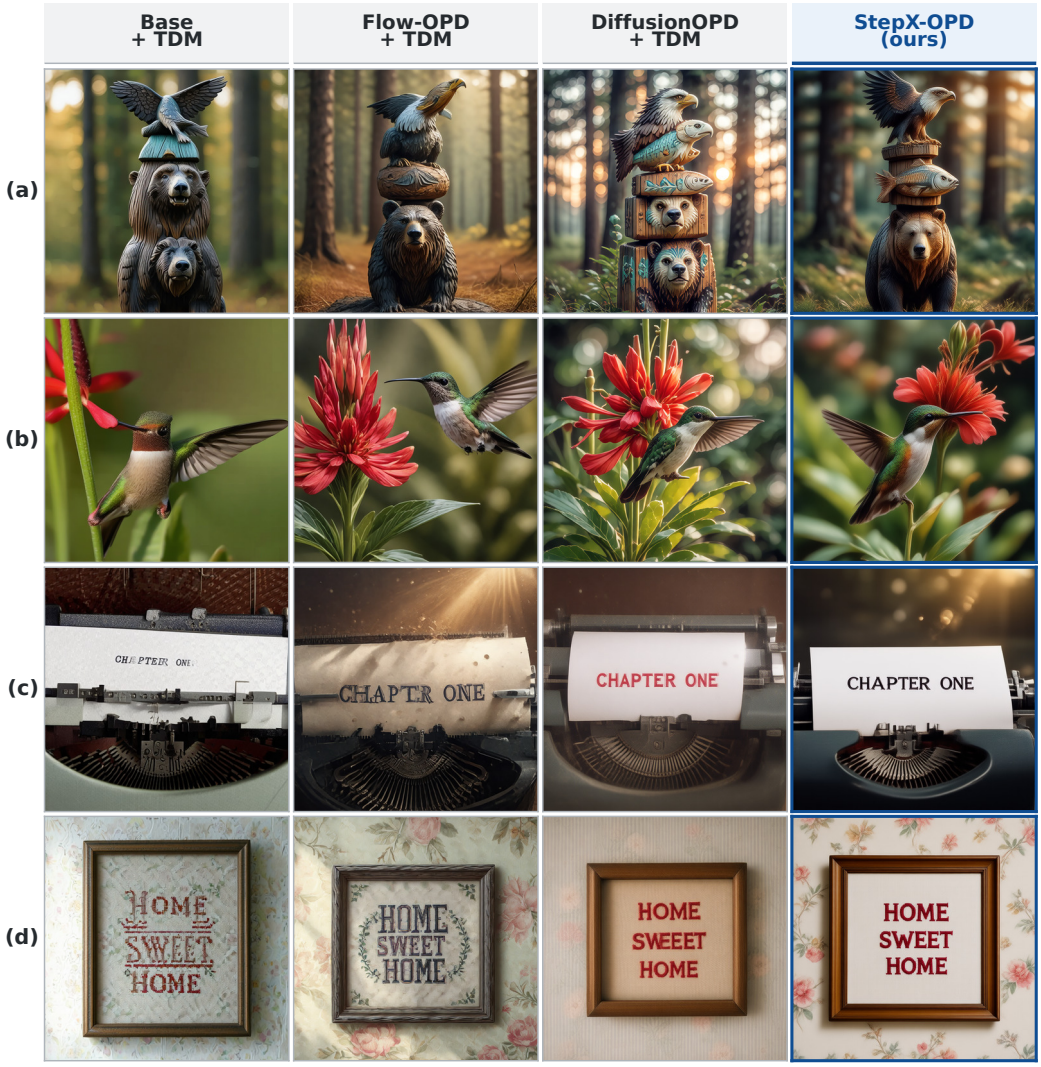


Figure 11: (a) Stacked wooden animal totem with bear at the bottom, eagle in the middle and fish on top, with weathered paint, forest bokeh and warm light; (b) A hummingbird hovering in front of a red trumpet flower with blurred green foliage behind, high-speed macro; (c) An old typewriter with a sheet of paper rolled in, the typed line reading "CHAPTER ONE", dust motes in a slanting sunbeam; (d) A hand-stitched sampler framed on a wall reading "HOME SWEET HOME", late afternoon light, faded floral wallpaper behind.

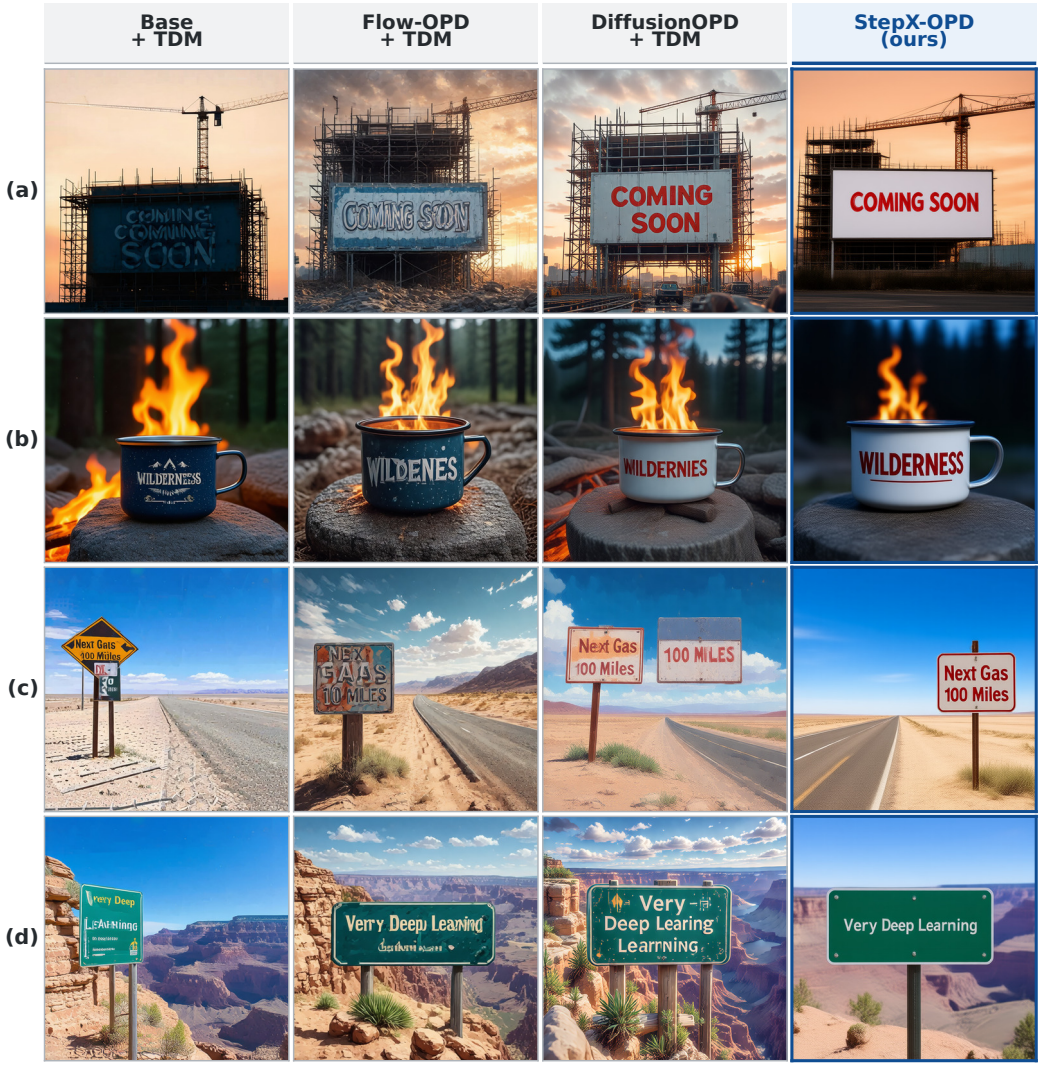


Figure 12: (a) A construction site hoarding painted with "COMING SOON", scaffolding above, crane silhouette against an orange evening sky; (b) An enamel mug on a campfire stone printed with "WILDERNESS", flames glowing, pine forest dark behind, blue hour; (c) A vast, sun-baked desert landscape with a lone highway stretching into the distance. A weathered road sign stands by the side, warning "Next Gas 100 Miles", under a blazing sky; (d) A green sign that says "Very Deep Learning" and is at the edge of the Grand Canyon.

G PRECISE TEACHER TARGETS AND PROMPT-WORD RECURRENCE

At the four-step working point, each sampling step spans a state interval far larger than a full-step sampler's, so a single instantaneous teacher velocity captures only one point of the local transition it should describe. The precise teacher target replaces that single evaluation with an average of guided velocities recomputed along a short rollout launched from the supervision state, so that one target covers the whole transition. It is an optional component, and Tab. 1 reports `StepX-OPD` without it. This appendix first gives its construction and then reports what it changes in the text these models write into images.

The construction starts from each supervision point z_τ and rolls out the active target velocity field v^{tgt}. Set $x^{(0)} = z_\tau$ and define

$$t_k = \frac{K-k}{K}\,\tau, \qquad k = 0, \dots, K, \tag{66}$$

so that $t_K = 0$. At each $k = 0, \dots, K-1$, evaluate the conditional and unconditional velocities at the same current state:

$$v_c^{\mathrm{tgt},(k)} = v^{\mathrm{tgt}}\big(x^{(k)}, t_k, c\big), \qquad v_u^{\mathrm{tgt},(k)} = v^{\mathrm{tgt}}\big(x^{(k)}, t_k, \varnothing\big), \tag{67}$$

Combine the two velocities with classifier-free guidance,

$$v_{\mathrm{cfg}}^{\mathrm{tgt},(k)} = v_u^{\mathrm{tgt},(k)} + w\,\big(v_c^{\mathrm{tgt},(k)} - v_u^{\mathrm{tgt},(k)}\big), \tag{68}$$

and use the explicit equal-interval Euler update

$$x^{(k+1)} = x^{(k)} + (t_{k+1} - t_k)\,v_{\mathrm{cfg}}^{\mathrm{tgt},(k)}, \tag{69}$$

recomputing both velocities at each new state. After K steps, define $\hat{x}_0 = x^{(K)}$ and average the guided velocities:

$$\bar{v}^{\mathrm{tgt}} = \frac{1}{K}\sum_{k=0}^{K-1} v_{\mathrm{cfg}}^{\mathrm{tgt},(k)}, \tag{70}$$

which requires $2K$ target-model forward evaluations in total.

Because the guided velocity is recomputed before every advance, the rollout follows the current guided direction at each step. With $\hat{x}_0 = x^{(K)}$, its average satisfies

$$z_\tau - \tau\,\bar{v}^{\mathrm{tgt}} = \hat{x}_0, \tag{71}$$

where $\hat{x}_0$ is the state reached at $t = 0$. Thus $\bar{v}^{\mathrm{tgt}}$ is a secant velocity, the average slope from z_τ to the guided trajectory's endpoint, rather than an instantaneous velocity at one point. Turning the precise teacher on makes both terms of Eq. 11 take $\bar{v}^{\mathrm{tgt}}$ in place of the instantaneous target $\bar{v}^{\mathrm{tgt}}$, which costs $2K - 2$ target-model forwards beyond the two that the instantaneous target uses.

Measured overhead. Turning the precise teacher on has a measured cost. It is observed on a controlled pair of runs that share every setting except the precise teacher, which is enabled only from step 1000 to step 1500 of training. Inside that window the per-step time has a median of 15.74 seconds without the precise teacher and 20.11 seconds with it, an overhead of 4.33 seconds per step. These figures come from the 460 paired ordinary training steps the window contains, leaving out the steps that absorb the cost of periodic evaluation and checkpoint writing. Outside the window the two runs show almost no difference, a mean per-step gap of 0.11 seconds, confirming that the added cost is confined to the window itself. Over the 500 steps the window spans, this overhead accumulates to about 36 minutes (2170 seconds). Because the window bounds the cost rather than letting it run for the training's full length, the accumulated cost does not grow as training continues; keeping the precise teacher on for the entire comparison instead of only the window would instead accumulate the same per-step overhead across the whole run, about 3.6 hours (3.64 h) in total. Measured against the cumulative time the ordinary training steps alone take to run, excluding the time spent on periodic evaluation and checkpoint writing, the window's cost is 4.8% of that time, and keeping the precise teacher on for the entire run would raise it to 28.9%.

Effect on prompt-word recurrence. Tab. 15 reports the same two counts as Tab. 12 and Tab. 13, on the same two prompt groups, for three four-step distillation configurations each evaluated with precise teacher supervision off and on. The first of these is a controlled pair, its two rows sharing a recipe and a training length and differing only in the precise teacher. On the group with no text

Table 15: Ablation of precise-teacher supervision across three distillation configurations, reporting text rendered, the count of generations with detected on-image text, and prompt-word recurrence, the count among those whose detected text reproduces a word from the prompt, on the prompt group with no text expected and, within it, on the further cut with no text called for, which additionally excludes prompts that name an existing entity or that describe an object able to carry text. Bold marks the lower prompt-word recurrence count within each configuration line.

			No text expected		No text called for	
Initialization	Teachers	Precise teacher	Text rendered	Prompt-word recurrence	Text rendered	Prompt-word recurrence
SFT-augmented merge	DiffusionOPD	Off	304	94	96	20
		On	298	**84**	92	**19**
Averaged-specialist merge	DiffusionOPD	Off	284	101	93	24
		On	292	**87**	86	**21**
Averaged-specialist merge	Flow-GRPO	Off	190	71	60	17
		On	236	**70**	71	**15**

Table 16: Ablation of soup-composition variants of the initialization, reported as full-benchmark scores.

Initialization	GenEval ↑	OCR ↑	DeQA ↑	PickScore ↑	Average ↑
Merge (3 teacher)	0.8709	0.8603	4.064	22.53	0.8527
Merge w/o OCR	0.8822	0.8253	4.078	22.62	0.8483
Merge w DEQA	0.8691	0.8346	4.086	22.58	0.8474

expected it moves from 94 to 84 on recurrence while its any-text count moves from 304 to 298, so recurrence falls by more than the count of generations carrying any text does. The other two configurations move the same way on recurrence, from 101 to 87 and from 71 to 70, with their any-text counts rising rather than falling; their two rows are read at different training lengths, so they record the direction without isolating the precise teacher. On the further cut with no text called for, the controlled pair moves from 20 to 19 on recurrence while its any-text count moves from 96 to 92, so there the recurrence count falls by less than the any-text count does. The any-text count is itself not uniform in direction: of the six cells this table reports, three are lower with precise teacher supervision on and three are higher.

H SOUP COMPOSITION AND VELOCITY-REGRESSION VARIANTS

Every cell in this section comes from a completed evaluation under the protocol in App. C, and fixed-prompt in-loop measurements are excluded.

Initialization definitions. Let W_0 denote the base weights, D_{GE}, D_{OCR}, and D_{Pick} the three specialist increments. The standard S3 initialization is

$$W_{\mathrm{S3}} = W_0 + \tfrac{1}{3}\big(D_{\mathrm{GE}} + D_{\mathrm{OCR}} + D_{\mathrm{Pick}}\big).$$

The no-OCR variant retains coefficient $1/2$ on each of the other two increments, $W_0 + \frac{1}{2}(D_{\mathrm{GE}} + D_{\mathrm{Pick}})$, without renormalization. The +DeQA variant is the four-source mean, $W_0 + \frac{1}{4}(D_{\mathrm{GE}} + D_{\mathrm{OCR}} + D_{\mathrm{Pick}} + D_{\mathrm{DeQA}})$. For the SFT variants, D_{SFT} is the step-2350 SFT increment: additive initialization is $W_0 + D_{\mathrm{SFT}} + \frac{1}{3}(D_{\mathrm{GE}} + D_{\mathrm{OCR}} + D_{\mathrm{Pick}})$, whereas the four-model mean is $W_0 + \frac{1}{4}(D_{\mathrm{SFT}} + D_{\mathrm{GE}} + D_{\mathrm{OCR}} + D_{\mathrm{Pick}})$. Tab. 16 uses low-noise velocity MSE from step 500 without a precise target, and its rows are aligned comparisons under the four-step full-benchmark protocol.

Soup composition. Tab. 16 compares which specialist increments are pooled into the initialization soup.

Table 17: Comparison of two velocity-regression query designs, both under full-range support, reported as full-benchmark scores. The two rows differ only in how the regression time is drawn.

vMSE query	GenEval ↑	OCR ↑	DeQA ↑	PickScore ↑	Average ↑
Continuous sampled τ	0.8664	0.7296	4.101	22.65	0.8219
Per-sample inference-grid point	0.8449	0.7123	4.098	22.57	0.8113

Velocity-regression evaluations. All rows below start velocity-MSE at step 500. A per-sample inference-grid query samples one of the four inference-grid points for each sample rather than evaluating all four points on every training step. Tab. 17 compares this per-sample inference-grid query against the continuous sampled τ query under full-range support, where the two rows differ only in the query design. Switching to the per-sample inference-grid query lowers Average from 0.8219 to 0.8113 and OCR from 0.7296 to 0.7123.