STEP: Post-Training Few-Step Flow Models by Propagating Reward Gradients Through Every Generation Step

Anonymous authors
Paper under double-blind review

Abstract

We study reward fine-tuning of few-step Gaussian-mixture (GM) flow policies. Each network evaluation defines a velocity field over a sampling interval, allowing analytic estimates of the clean latent at intermediate states. We introduce STEP, which uses these estimates as probes for constructing surrogate reward gradients. One probe is sampled within each deployment interval, and a straight-through chain retains dependencies between intervals. Weighted fusion connects all probes to a shared terminal reward while preserving the sampled image in the forward pass. The probes require no additional transformer evaluations. The full training objective combines this terminal loss with auxiliary rewards and frozen-teacher velocity regularization. After 500 optimization steps on FLUX.1-dev, STEP achieves 0.530 GenEval, 0.580 OCR, and 15.80 held-out HPSv3 with four neural function evaluations. The official distilled student scores 0.396, 0.577, and 13.47, respectively. Within the full objective, removing the chain or replacing interior probes with zero-depth estimates reduces OCR and GenEval. These results support using the interval structure of GM policies for reward fine-tuning at a fixed four-step deployment budget.

1 Introduction

Text-to-image models must produce images that satisfy user preferences while remaining efficient to sample. Flow-matching models generate high-quality images through iterative denoising (Lipman et al., 2023; Liu et al., 2023; Black Forest Labs, 2024). Distillation reduces this cost to a few neural function evaluations (NFEs) (Salimans & Ho, 2022; Song et al., 2023; Yin et al., 2024b; Chen et al., 2026), while reward fine-tuning improves properties not directly optimized during pretraining (Xu et al., 2023; Black et al., 2024; Ma et al., 2025). Preserving reward improvements when reducing the sampling budget is a separate challenge (Jiang et al., 2026). We therefore study reward fine-tuning directly at the few-step budget used for deployment.

We study how to assign gradients from a terminal image reward to the denoising steps of a few-step policy. Policy-gradient methods estimate updates from sampled trajectories (Black et al., 2024; Fan et al., 2023; Liu et al., 2025). Direct-gradient methods instead differentiate a reward through the sampling process (Prabhudesai et al., 2023; Clark et al., 2024). Surrogate methods separate the sampled trajectory from the graph used for differentiation. LeapAlign preserves sampled states through straight-through connectors and attenuates cross-step gradients (Liang et al., 2026). FlowBP studies the choice of active steps, integration weights, and cross-step coupling for a shared terminal reward (Wang et al., 2026). These methods motivate examining how the internal structure of a few-step policy can inform the surrogate gradient.

Gaussian-mixture (GM) flow policies provide the structure needed for this construction (Chen et al., 2025b; 2026). One transformer evaluation predicts a mixture over clean latents. Its analytic posterior defines a velocity field within the corresponding schedule interval. Numerical substeps reuse that field, and a clean-latent estimate can be evaluated at an interior depth without another transformer evaluation. We use these estimates as *probes*: differentiable clean-latent estimates that connect interval computations to the terminal reward. This gives a gradient construction based on the GM posterior within each interval.

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.

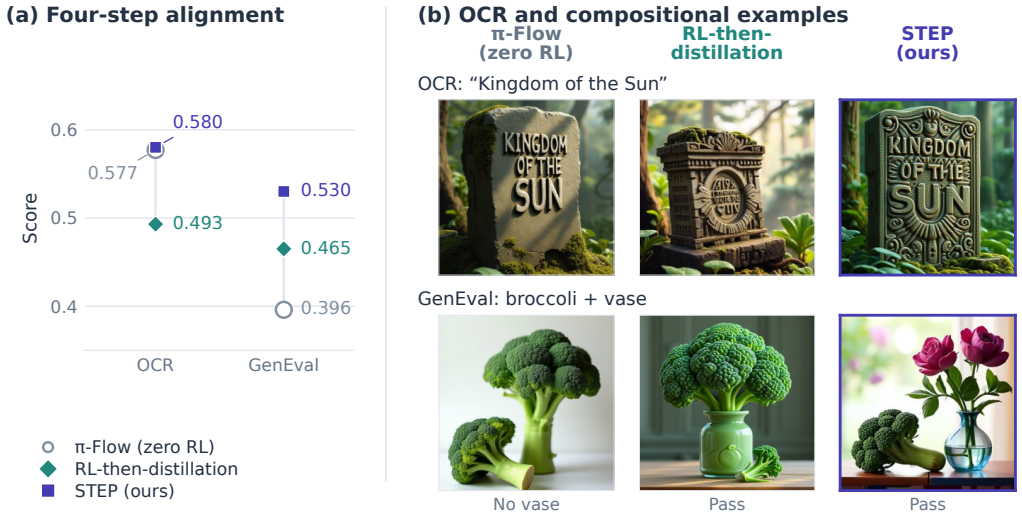

Figure 1: **Reward fine-tuning with four deployment steps.** (a) STEP achieves higher GenEval than the official distilled student and higher OCR than RL followed by distillation. (b) Examples from the OCR and GenEval benchmarks. All three models use four NFEs at inference. STEP is initialized from the base model and uses a frozen teacher for regularization. STEP scores are measured at optimizer step 500; the examples come from separate qualitative evaluations (Appendix A.5).

We introduce STEP (**S**hared-**T**erminal **E**valuation of intra-step **P**robes). STEP first stores a deployment trajectory without gradient tracking. It then recomputes the schedule heads, connects them through a straight-through chain (Bengio et al., 2013; Liang et al., 2026), and constructs one random-depth probe per interval. Weighted fusion combines the probes while preserving the sampled terminal latent in the forward pass. A single decoder and reward evaluation can therefore supply the latent gradient shared by all probes. The full objective also supervises intermediate estimates with auxiliary rewards and regularizes velocities with a frozen four-step teacher. The student is initialized from the base GM conversion and undergoes no separate distillation stage.

On FLUX.1-dev at four NFEs, STEP reaches 0.530 GenEval and 15.80 held-out HPSv3 after 500 optimization steps, exceeding the official distilled student on both metrics. OCR is comparable (0.580 vs. 0.577; Figure 1). The adapted LeapAlign baseline has higher GenEval, while STEP has higher OCR and HPSv3 in the reported comparison. Component ablations favor retaining the chain and probes from all intervals within the full objective. Removing auxiliary supervision also reduces performance, showing that the results depend on the combined training objective.

Our main contribution is a surrogate gradient construction that combines analytic probes inside GM intervals with fusion across all deployment steps. We specify its forward and backward computation and distinguish the relative probe weights from their overall scale. We evaluate this construction at four NFEs through comparisons with adapted baselines, component ablations, and gradient diagnostics.

2 Related Work

Reward optimization. DDPO and DPOK apply policy gradients to denoising (Black et al., 2024; Fan et al., 2023). Flow-GRPO and related methods extend group-relative optimization to flow matching (Liu et al., 2025; Xue et al., 2025b; Li et al., 2025a). Temporal variants use intermediate rewards, advantages, or branches to improve credit assignment (Savani et al., 2026; Liao et al., 2025; He et al., 2025; Li et al., 2025b; Zhou et al., 2025). Other approaches optimize the forward process (Zheng et al., 2026; Xue et al., 2025a) or learn from preference pairs (Wallace et al., 2024; Yang et al., 2024; Liang et al., 2025). STEP requires a differentiable reward and uses its gradient directly.

Direct reward gradients. ReFL, AlignProp, and DRaFT differentiate image rewards through all or part of a sampler (Xu et al., 2023; Prabhudesai et al., 2023; Clark et al., 2024). DRTune stops gradients

through model inputs (Wu et al., 2024), while adjoint objectives provide an alternative formulation for memory-efficient optimization (Domingo-Enrich et al., 2025). LeapAlign combines two-step leap trajectories with straight-through connectors, attenuated nested gradients, and trajectory-distance weighting (Liang et al., 2026). FlowBP organizes surrogate gradients by reward input, active steps, integration weights, and bridge coupling (Wang et al., 2026). STEP adopts the distinction between sampled values and surrogate gradient paths. Its specific construction uses analytic GM posterior means at interior states as probes, then fuses their gradient paths across all deployment intervals. The main reward is evaluated on the sampled terminal image.

Alignment of few-step generators. Distillation reduces sampling cost (Salimans & Ho, 2022; Song et al., 2023; Luo et al., 2023; Yin et al., 2024b;a; Sauer et al., 2024). GMFlow and π-Flow provide the mixture parameterization and analytic posterior updates used here (Chen et al., 2025b; 2026). Few-step alignment includes fine-tuning distilled students (Miao et al., 2025; Zhang et al., 2026), combining distillation with reward optimization (Jiang et al., 2026; Chen et al., 2025a; Huang et al., 2026), and using learned surrogate rewards for non-differentiable objectives (Luo et al., 2026). STEP studies reward-gradient construction within an existing GM policy family. It initializes the student from the base GM conversion and optimizes it at four NFEs. A distilled teacher provides velocity regularization during training.

3 Method

3.1 Problem Setup and GM Policy

Let y denote a text prompt and $x \in \mathbb{R}^{C\times H\times W}$ a latent, with channel, height, and width dimensions C, H, and W. An N-step sampler maps $x_{\text{init}} \sim \mathcal{N}(0, I)$ to a terminal latent x_f. Interval i spans noise levels $(\sigma_i^{\text{src}}, \sigma_i^{\text{dst}})$ and requires one transformer evaluation. We refer to this evaluation and the subsequent integration as a *schedule head*. The transformer parameters are shared across heads. We use $N = 4$ in our experiments.

The student uses FLUX.1-dev with a GM output parameterization (Black Forest Labs, 2024), initialized from the base checkpoint without a distilled adapter. Its frozen output projection predicts $K = 8$ mixture components, and only LoRA weights are trained (Hu et al., 2022). We write θ for these trainable parameters and omit conditioning on y in the policy notation. Given a frozen VAE decoder D and a frozen differentiable reward r_ϕ, we aim to increase $\mathbb{E}[r_\phi(D(x_f), y)]$ under a fixed sampling budget. We use HPSv3 during training (Ma et al., 2025).

A GM head predicts a distribution $q_{\theta,i}(z \mid x_i)$ over clean latents at its source state x_i (Chen et al., 2025b; 2026). At a state (x_t, σ_t) inside that interval, the analytic update is

$$\begin{aligned} q_{\theta,i}(z \mid x_t, x_i) &\propto \frac{p_t(x_t \mid z)}{p_i(x_i \mid z)} q_{\theta,i}(z \mid x_i), \\ \bar{x}_{0,\theta,i}(x_t, \sigma_t) = \mathbb{E}[z \mid x_t, x_i], &\qquad \pi_{\theta,i}(x_t, \sigma_t) = \frac{x_t - \bar{x}_{0,\theta,i}(x_t, \sigma_t)}{\max(\sigma_t, \varepsilon)}. \end{aligned} \tag{1}$$

Here $p_t(x_t \mid z) = \mathcal{N}(x_t; (1-\sigma_t)z, \sigma_t^2 I)$; p_i uses σ_i^{src}, and $\varepsilon > 0$ prevents division by zero. The subscript 0 in $\bar{x}_{0,\theta,i}$ denotes a clean-latent estimate. Appendix A.2 gives the mixture parameterization. Euler substeps reuse this velocity field without further transformer evaluations. The sampler and its stochastic settings are specified in Appendix A.3.

3.2 Surrogate Gradients across the Sampling Schedule

Cached trajectory and straight-through chain. We store the deployment states x_i and terminal latent x_f without gradient tracking, then recompute all schedule heads with gradients enabled. For $i = 0, \ldots, N-1$, let $F_{\theta,i}(t, a)$ denote integration over interval i, with transformer input latent t and initial state a. With $a_0 = x_0 = x_{\text{init}}$ and $x_N = x_f$, define

$$\begin{aligned} t_i &= \rho a_i + (1-\rho)\,\text{sg}(a_i), \\ \widehat{x}_{i+1} &= F_{\theta,i}(t_i, a_i), \\ a_{i+1} &= \widehat{x}_{i+1} + \text{sg}(x_{i+1} - \widehat{x}_{i+1}). \end{aligned} \tag{2}$$

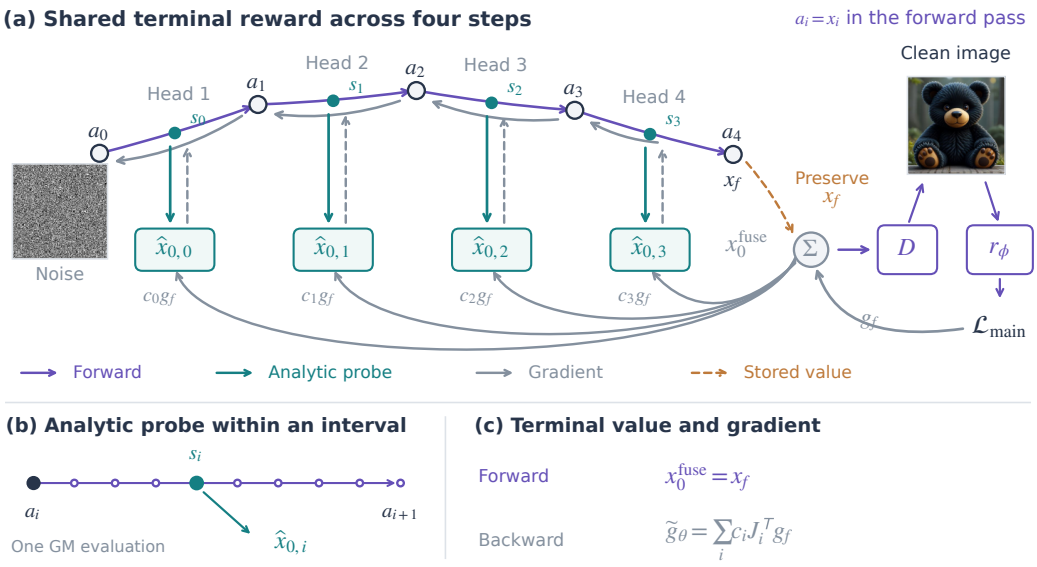

Figure 2: **STEP: four-step training with a shared terminal reward.** **(a)** Purple paths retain cached states $a_i = x_i$; gray paths differentiate the recomputed straight-through chain (equation 2). Orange supplies the detached terminal latent x_f. D and r_ϕ are the frozen decoder and reward model; $\mathcal{L}_{\text{main}}$ is the terminal hinge loss. **(b)** Head $i+1$ ($i = 0, \ldots, 3$) computes clean-latent probe $\widehat{x}_{0,i}$ at sampled substep s_i using one GM prediction (equation 3). **(c)** Fusion keeps $x_0^{\text{fuse}} = x_f$ in the forward pass (equation 4). Here g_f is the latent gradient of $\mathcal{L}_{\text{main}}$ at x_0^{fuse}; c_i is a detached fusion weight; and J_i is the probe Jacobian with respect to LoRA parameters θ, including the chain. The weighted contributions give the surrogate parameter gradient $\widetilde{g}_\theta$ (equation 6). Noise is schematic; the clean image is an existing STEP output. Auxiliary losses and teacher regularization are described in Section 3.3.

Following LeapAlign (Liang et al., 2026), we set $\rho = 0.3$ to attenuate the gradient through each transformer input, with $t_0 = a_0$. The operator sg stops gradients. Forward values satisfy $a_{i+1} = x_{i+1}$, while gradients follow the recomputed trajectory. Integration states remain differentiable across all N deployment intervals, with one recomputed GM head per interval.

Random-depth probes. Head i uses n_i numerical substeps from the sampler's total budget of 130. We sample $s_i \sim \text{Unif}\{1, \ldots, n_i - 1\}$ and evaluate the posterior mean after the first s_i substeps:

$$\begin{aligned} x_{i,s_i} &= F_{\theta,i}^{s_i}(t_i, a_i), \\ \widehat{x}_{0,i} &= \bar{x}_{0,\theta,i}(x_{i,s_i}, \sigma_{i,s_i}), \\ d_i &= \text{sg}\left[\max\left(\tau, \frac{\|\widehat{x}_{0,i} - x_f\|_1}{CHW}\right)\right], \qquad \tau = 0.1. \end{aligned} \tag{3}$$

Each probe reuses the GM prediction for its interval. The lower bound τ on the distance prevents arbitrarily large inverse-distance weights for probes close to x_f.

Probe fusion for a shared terminal reward. Evaluating a separate reward for every probe would require N decoder and reward evaluations. We instead adapt trajectory-distance weighting (Liang et al., 2026) to fuse the GM probes:

$$c_i = \frac{d_i^{-1}}{(\sum_j d_j)(\sum_k d_k^{-1})}, \qquad x_0^{\text{fuse}} = \text{sg}(x_f) + \sum_{i=0}^{N-1} c_i \big(\widehat{x}_{0,i} - \text{sg}(\widehat{x}_{0,i})\big). \tag{4}$$

The distances d_i and coefficients c_i are constant in the backward pass. The forward value is exactly x_f, and probe i receives the terminal latent gradient scaled by c_i. Writing $c_i = \gamma w_i$, where $w_i = d_i^{-1}/\sum_k d_k^{-1}$ and $\gamma = 1/\sum_j d_j$, separates two effects. The relative weights w_i favor probes closer to x_f; the common factor γ sets their total scale for each sample. The sum of the coefficients

satisfies

$$\sum_{i=0}^{N-1} c_i = \frac{1}{\sum_i d_i} \le \frac{1}{N\tau}. \tag{5}$$

This is a bound on the coefficients. The magnitude and direction of the parameter gradient also depend on the probe and reward Jacobians. Appendix C.1 provides the derivation.

To express the resulting gradient, let $g_f = \nabla_{x_0^{\text{fuse}}} \mathcal{L}_{\text{main}}$ denote the latent gradient of the terminal loss. Let J_i be the Jacobian of probe $\widehat{x}_{0,i}$ with respect to the trainable parameters, computed through the straight-through chain. The terminal loss yields

$$\widetilde{g}_\theta = \sum_{i=0}^{N-1} c_i J_i^\top g_f. \tag{6}$$

The probes share the same terminal latent gradient g_f, while each J_i accounts for its dependence on the corresponding head and the chain.

Equations 2–4 specify the straight-through computation. The forward pass preserves the stored states and terminal image, while the backward pass follows the surrogate gradient paths. This generally differs from the gradient obtained by differentiating the deployment sampler.

3.3 Training Objective and Cost

The training objective combines terminal and auxiliary reward losses with teacher regularization. The terminal term uses a hinge loss:

$$\mathcal{L}_{\text{main}} = g\ \text{ReLU}\Big(M - r_\phi(D(x_0^{\text{fuse}}), y)\Big), \qquad (g, M) = (0.1, 11). \tag{7}$$

Only samples below the margin can contribute a reward gradient through this term. Its gradient is zero for samples above the margin. At update s, head $i^\star = s \bmod N$ receives auxiliary supervision at interval fractions $\mathcal{F} = \{0.6, 0.8\}$:

$$\mathcal{L}_{\text{aux}} = g \sum_{f \in \mathcal{F}} \lambda_f\, \text{ReLU}\big(M - r_\phi(D(\widehat{x}_{0,i^\star,f}), y)\big), \qquad (\lambda_{0.6}, \lambda_{0.8}) = (0.3, 0.7). \tag{8}$$

The intermediate estimates use the posterior mean in equation 3 at the specified interval fractions. We process the two auxiliary decoder and reward graphs sequentially, releasing each after computing its latent gradient. A frozen four-step π-Flow teacher defines the velocity regularizer (teacher anchor)

$$\mathcal{L}_{\text{reg}} = \frac{1}{NCHW} \sum_{i=0}^{N-1} \left\| \pi_{\theta,i}(a_i, \sigma_i^{\text{src}}) - \pi_{\text{T},i}(\text{sg}(a_i), \sigma_i^{\text{src}}) \right\|_2^2. \tag{9}$$

The teacher is used only for this regularization term. Starting from the base-model initialization, we optimize

$$\mathcal{L}(\theta) = \mathbb{E}\big[\mathcal{L}_{\text{main}} + \omega_s \mathcal{L}_{\text{aux}} + \beta \mathcal{L}_{\text{reg}}\big], \quad \omega_s = \omega \mathbf{1}[s \ge 50], \quad (\omega, \beta) = (0.1, 0.05). \tag{10}$$

The expectation is over prompts, initial noise, sampler randomness, and probe depths. Setting $\omega = 0$ removes the auxiliary reward; setting $\beta = 0$ removes teacher regularization.

Each training trajectory requires N student evaluations without gradient tracking, N student evaluations with gradients enabled, and N frozen teacher evaluations. The terminal loss requires one decoder and reward evaluation; the auxiliary loss adds two. The analytic probes require no additional transformer evaluations. Inference uses $N = 4$ transformer evaluations. Section 4.6 reports training times and the number of images evaluated by the reward model.

4 Experiments

4.1 Setup

Training. We use FLUX.1-dev (Black Forest Labs, 2024) with a frozen GM output projection, $K = 8$ components, and rank-32 LoRA (Hu et al., 2022). We train for 500 optimizer steps with

Table 1: **Four-step evaluation on FLUX.1-dev.** All rows use our evaluation prompts and scorers. The zero-RL reference is distilled; the other rows start from the base GM conversion. Reward-trained models are evaluated at optimizer step 500; training costs differ (Section 4.6). HPSv3† is the training scorer. PS denotes raw PickScore logits, and IR denotes ImageReward. Higher is better in every column.

	Verifiable alignment		Training scorer	Unseen scorers		
Method	GenEval	OCR	HPSv3†	HPSv2.1	PS	IR
π-Flow (zero RL)	0.396	0.577	13.47	0.300	22.5	0.988
DRaFT	0.460	0.480	14.20	0.330	22.7	1.250
LeapAlign+substep	**0.550**	0.560	15.50	0.360	22.8	1.280
Flow-GRPO	0.500	0.550	15.10	0.350	**23.1**	**1.360**
STEP (ours)	0.530	**0.580**	**15.80**	**0.380**	23.0	1.350

AdamW (Loshchilov & Hutter, 2019), a learning rate of 10^{-4}, and 64 trajectories per step. HPSv3 (Ma et al., 2025) provides the training reward, and prompts are sampled from HPDv2 (Wu et al., 2023). The deployment sampler uses 512^2 images, guidance 3.5, four transformer evaluations, scheduler shift 3.2, final-step scale 0.5, and 130 analytic substeps. Appendix A provides the full configuration.

Evaluation. We evaluate composition on GenEval (Ghosh et al., 2023) (553 prompts, four images each) and text rendering on the 1018-prompt OCR benchmark using the scorer of Liu et al. (2025). We evaluate reward scores on 400 held-out HPDv2 prompts with four images each. HPSv3 is used for training; HPSv2.1 (Wu et al., 2023), PickScore (Kirstain et al., 2023), ImageReward (Xu et al., 2023), and UnifiedReward-Align/IQ (Wang et al., 2025) are used only for evaluation. The LAION aesthetic predictor (Schuhmann, 2022) and DeQA-Score (You et al., 2025) measure image quality on these prompts and on 200 DrawBench prompts (Saharia et al., 2022), with four images each. Training seeds are 42/43/44 where multiple runs are available. STEP, its component ablations, and the three adapted baselines are evaluated at optimizer step 500. The zero-RL student and two-stage routes use the reference checkpoints described in Appendix A.5.

Comparisons. We adapt DRaFT (Clark et al., 2024), LeapAlign+substep (Liang et al., 2026), and Flow-GRPO (Liu et al., 2025) to the same backbone, training reward, prompt distribution, and four-NFE sampler. Loss terms and computation per update differ across methods (Appendix A.8). The official distilled π-Flow student (Chen et al., 2026) is a zero-RL reference, the initialization for distill-then-RL, and the frozen teacher used for regularization. STEP is initialized from the base GM conversion. RL-then-distillation first optimizes the reward at 16 NFEs and then distills the model to four steps. Appendix B.6 reports published results under the original evaluation protocols.

4.2 Four-Step Performance

Table 1 compares the four-step models at optimizer step 500. STEP has the highest reported OCR, HPSv3, and HPSv2.1 scores among the adapted baselines. It exceeds DRaFT by 7.0 percentage points on GenEval and 10.0 on OCR. These results describe the full training configurations; Section 4.4 examines components within the STEP objective.

Relative to the distilled student, STEP raises GenEval from 0.396 to 0.530 and HPSv3 from 13.47 to 15.80, with comparable OCR (0.580 vs. 0.577). STEP uses HPSv3 rewards and teacher regularization, without task-specific OCR or GenEval rewards. LeapAlign+substep retains higher GenEval (0.550), while Flow-GRPO has slightly higher PickScore (23.1 vs. 23.0) and ImageReward (1.360 vs. 1.350).

4.3 Comparison with Two-Stage Training

Table 2 compares direct four-step training with two-stage procedures. For RL-then-distillation, OCR falls from 0.590 at 16 NFEs to 0.493 after distillation. Deploying the 16-NFE teacher directly at four steps also gives lower GenEval than its distilled student (0.436 vs. 0.465). These differences describe the particular training and distillation runs in this comparison. STEP evaluates rewards on the four-step output and reaches 0.580 OCR and 0.530 GenEval without subsequent distillation.

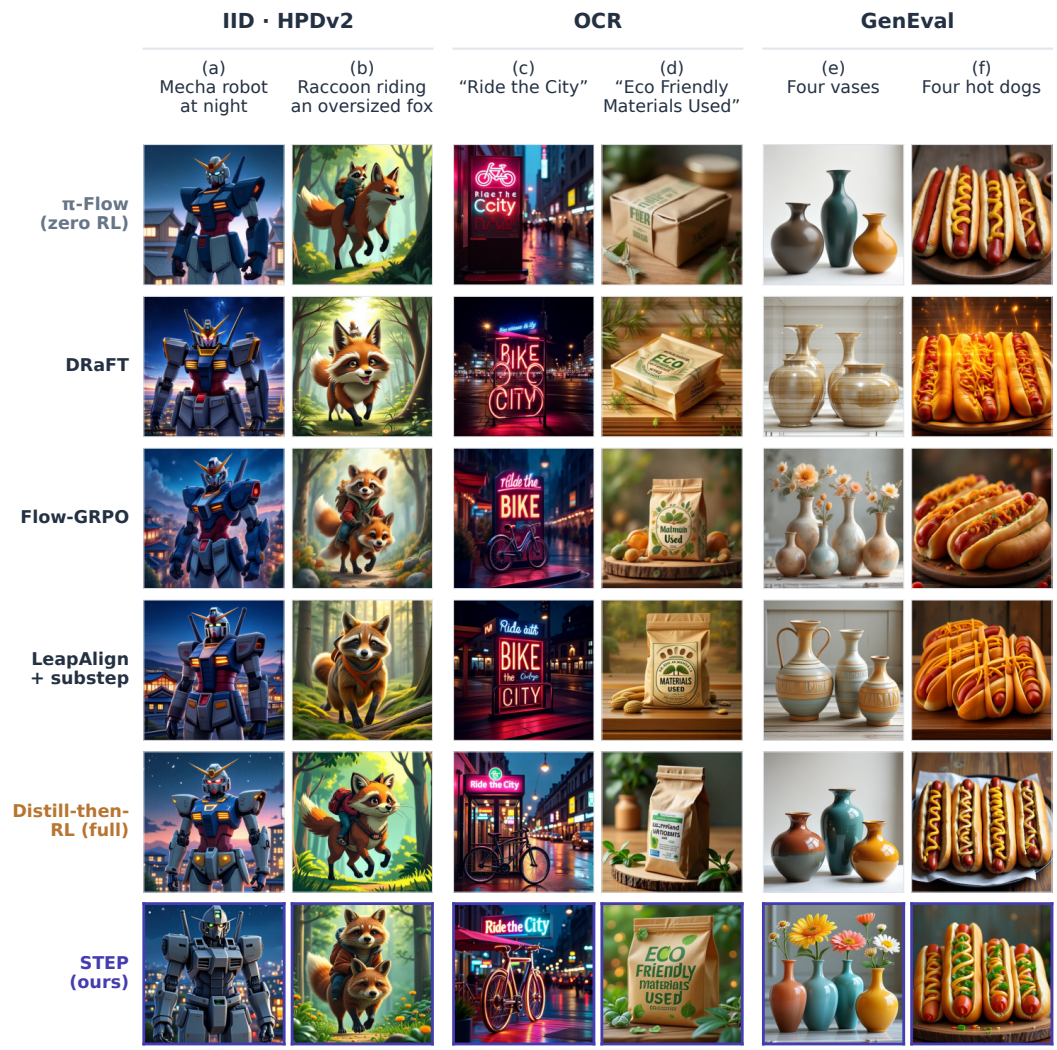

Figure 3: **Qualitative comparison across three prompt types.** Columns show two HPDv2, two OCR, and two GenEval examples; rows compare six methods. Prompt excerpts appear above the images; Table 13 gives the full prompts and generation indices. Distill-then-RL uses the full training objective. Examples are selected for visible differences in image quality, text rendering, or composition. Images come from separate qualitative evaluations; Table 1 reports the 500-step quantitative evaluation. Appendix D provides additional examples with different prompts.

Distill-then-RL begins with the distilled four-step student and achieves 0.587 OCR, close to STEP's 0.580, with lower GenEval (0.413 vs. 0.530). The routes differ in initialization and training history, so this comparison does not isolate the order of reward fine-tuning and distillation. Initialization can also affect hinge activity: the distilled student's initial mean validation reward exceeds the margin (Appendix C.2). We interpret these results as comparisons of the specified training routes.

4.4 Component Ablations

Cross-step dependencies and intra-step probes. Removing the chain lowers GenEval from 0.530 to 0.490 and OCR from 0.580 to 0.520. Restricting fusion to a single probe produces a larger GenEval reduction, to 0.420. Replacing intra-step probes with zero-depth probes also reduces GenEval to 0.480 and OCR to 0.520. These results favor the full chain and interior probes across the schedule under the combined objective. The zero-depth comparison tests interior estimates; it does not distinguish random depths from fixed interior depths.

Table 2: **Training routes to a four-step model.** The zero-RL student initializes the distill-then-RL variants. STEP is initialized from the base GM conversion. All models use four NFEs at inference; training procedures and checkpoints are specified in Appendix A.5.

Training route	OCR ↑	GenEval ↑
π-Flow (zero RL)	0.577	0.396
RL-then-distillation	0.493	0.465
Distill-then-RL (full objective)	0.587	0.413
Aux only, no anchor	0.541	0.391
No aux, no anchor	0.499	0.414
STEP: direct training at four NFEs	0.580	0.530

Table 3: **Component ablations at optimizer step** 500**.** All variants use the same four-NFE sampler. Removing three probes leaves a single probe (ChainB). Mean OCR scores across training seeds appear separately in Table 4.

Variant	OCR ↑	GenEval ↑
STEP full	0.580	0.530
– auxiliary reward	0.500	0.450
– teacher anchor	0.600	0.520
– chain	0.520	0.490
– intra-step probes	0.520	0.480
– three probes	0.520	0.420

Auxiliary supervision and teacher regularization. Removing the auxiliary reward reduces both OCR and GenEval by 0.080, showing the importance of intermediate supervision in the full objective. Auxiliary gradients can remain active when a terminal reward exceeds the hinge margin, provided that the intermediate rewards remain below it (Appendix C.2). Removing teacher regularization increases OCR to 0.600 and lowers GenEval to 0.520 in the reported comparison; the two variants have the same mean OCR across training seeds (Table 4). Figure 4 illustrates changes in subject arrangement and expression.

Appendix C.3 examines gradient propagation through the schedule heads (Figure 6). From the first head to the last, mean probe distance decreases from 0.418 to 0.100, while the fusion coefficient increases from 0.113 to 0.468. The later probes are closer to the terminal latent and receive larger coefficients by construction. Under the unsaturated diagnostic loss, all four heads receive nonzero gradients, with mean norms from 3.47 to 11.18. This verifies connectivity of the surrogate paths under that loss. Their activity under the training hinge depends on the sample rewards; these norms do not measure the contribution of each loss term to the final model.

4.5 Generalization across Metrics and Prompt Sets

STEP also improves metrics not used for training (Table 6). Relative to the zero-RL student, HPSv2.1 rises from 0.300 to 0.380, ImageReward from 0.988 to 1.350, and UnifiedReward-Align from 0.674 to 0.720. Aesthetic and DeQA improve on both HPDv2 and DrawBench (Table 7); for example, DrawBench scores rise from 5.61/4.33 to 6.00/4.40. The DrawBench results extend this comparison to a different prompt set. Together, these measurements show that the full objective improves several evaluation metrics beyond its training reward, relative to the distilled reference.

4.6 Training Cost

Probe fusion uses one terminal decoder and reward evaluation, with no additional transformer evaluations for the probes. Two auxiliary rewards increase the total to at most 192 scored images per optimizer step, compared with 64 for DRaFT and Flow-GRPO. Mean recorded step times are 50.3 s for STEP, 39.2 s for DRaFT, and 19.5 s for Flow-GRPO. These averages use different log windows (Appendix B.5). The 500-step comparison matches trajectory counts rather than total computation. Additional reward and teacher evaluations occur only during training; inference uses four NFEs.

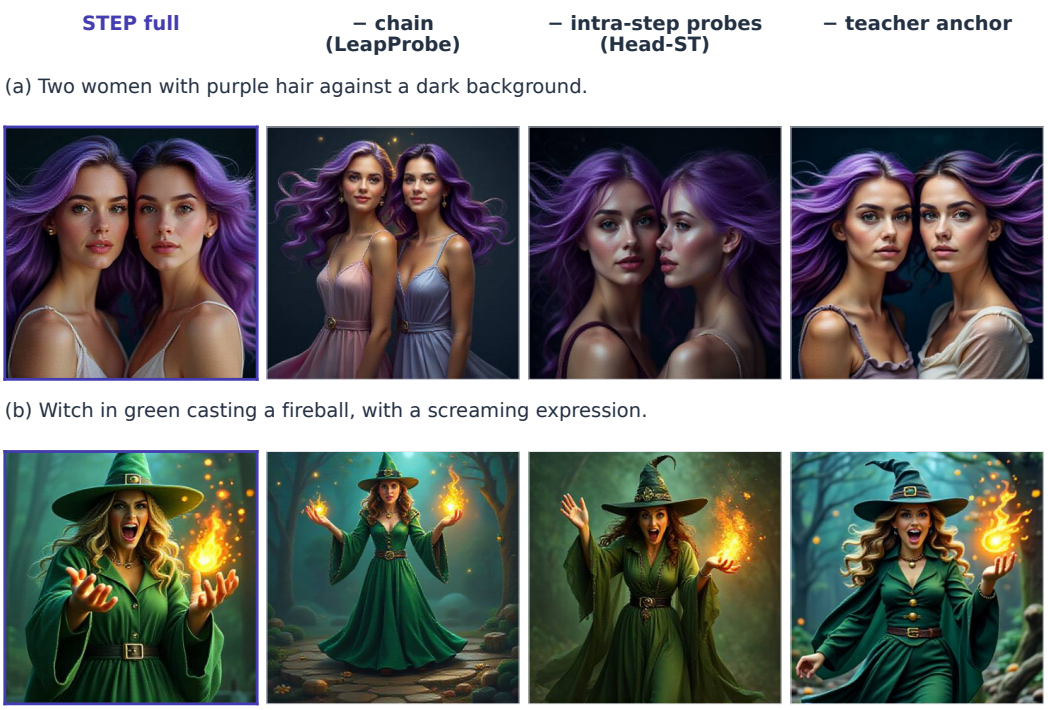

Figure 4: **Qualitative comparison of component ablations.** Each row uses the same HPDv2 prompt across four variants. The examples show changes in subject arrangement and facial expression. Head-ST replaces intra-step probes with zero-depth probes; LeapProbe removes the chain. All four variants use checkpoint 500, matching the quantitative ablations in Table 3. Table 14 provides the full prompts. The examples are not repeated in the supplementary figures.

5 Conclusion

STEP constructs surrogate reward gradients from analytic probes inside GM sampling intervals. A straight-through chain retains cross-step dependencies, and fusion connects probes from all deployment intervals to a shared terminal reward. The full objective combines this signal with intermediate rewards and teacher regularization. At four NFEs and 500 optimization steps, STEP reaches 0.530 GenEval, 0.580 OCR, and 15.80 held-out HPSv3. The component comparisons favor the full chain and interior probes within this objective and show a substantial contribution from auxiliary supervision. Evaluation on additional scorers and DrawBench finds improvements over the distilled reference beyond the training reward.

Limitations and future work. Our study is limited to FLUX.1-dev with four-step sampling and LoRA updates. We plan to evaluate additional backbones, sampling budgets, and independent training seeds. The baseline comparisons match optimizer steps but use different objectives and costs. Future comparisons will match auxiliary supervision, teacher regularization, and GPU budgets. We will also measure hinge activity and the gradient contributions of individual loss terms. Fixed-depth and matched-scale controls will examine probe sampling and relative weighting. Finally, STEP uses a biased surrogate for GM policies with differentiable rewards. We plan to compare its update directions with gradients of the deployment sampler and measure the resulting changes in terminal reward.

AI Use Statement

AI tools were used to improve the language and clarity of the manuscript.

Reproducibility Statement

Section 3 defines the training surrogate. Section 4.1 and Appendix A specify the model, optimization, sampler, data, scorers, seeds, and evaluation checkpoints. Appendix B reports the evaluation tables, and Appendix C analyzes the gradient computation. Training code, configuration files, and evaluation scripts will be released upon publication.

References

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation. *arXiv preprint arXiv:1308.3432*, 2013. URL https://arxiv.org/abs/1308.3432.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training Diffusion Models with Reinforcement Learning. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/14f75513f0f1ca01de1e826b52e6b840-Abstract-Conference.html.

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

Guanjie Chen, Shirui Huang, Kai Liu, Jianchen Zhu, Xiaoye Qu, Peng Chen, Yu Cheng, and Yifu Sun. Flash-DMD: Towards High-Fidelity Few-Step Image Generation with Efficient Distillation and Joint Reinforcement Learning. *arXiv preprint arXiv:2511.20549*, 2025a. URL https://arxiv.org/abs/2511.20549.

Hansheng Chen, Kai Zhang, Hao Tan, Zexiang Xu, Fujun Luan, Leonidas Guibas, Gordon Wetzstein, and Sai Bi. Gaussian Mixture Flow Matching Models. In *International Conference on Machine Learning*, volume 267, pp. 9783–9802, 2025b. URL https://proceedings.mlr.press/v267/chen25cl.html.

Hansheng Chen, Kai Zhang, Hao Tan, Leonidas Guibas, Gordon Wetzstein, and Sai Bi. π-Flow: Policy-Based Few-Step Generation via Imitation Distillation. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/f593c9c251d4d7cf14d4ab9861dfb7eb-Abstract-Conference.html.

Kevin Clark, Paul Vicol, Kevin Swersky, and David J Fleet. Directly Fine-Tuning Diffusion Models on Differentiable Rewards. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=1vmSEVL19f.

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. *arXiv preprint arXiv:2507.05595*, 2025. URL https://arxiv.org/abs/2507.05595.

Carles Domingo-Enrich, Michal Drozdzal, Brian Karrer, and Ricky T. Q. Chen. Adjoint Matching: Fine-tuning Flow and Diffusion Generative Models with Memoryless Stochastic Optimal Control. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2409.08861.

Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. DPOK: Reinforcement Learning for Fine-tuning Text-to-Image Diffusion Models. In *Advances in Neural Information Processing Systems*, 2023. doi: 10.52202/075280-3497. URL https://doi.org/10.52202/075280-3497.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. GenEval: An Object-Focused Framework for Evaluating Text-to-Image Alignment. In *Advances in Neural Information Processing Systems*, 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/a3bf71c7c63f0c3bcb7ff67c67b1e7b1-Abstract-Datasets_and_Benchmarks.html.

Xiaoxuan He, Siming Fu, Yuke Zhao, Wanli Li, Jian Yang, Dacheng Yin, Fengyun Rao, and Bo Zhang. TempFlow-GRPO: When Timing Matters for GRPO in Flow Models. *arXiv preprint arXiv:2508.04324*, 2025. URL https://arxiv.org/abs/2508.04324.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2106.09685.

Yushi Huang, Xiangxin Zhou, Ruoyu Wang, Chi Zhang, Jun Zhang, and Tianyu Pang. Reinforcing Few-step Generators via Reward-Tilted Distribution Matching. *arXiv preprint arXiv:2605.26108*, 2026. URL https://arxiv.org/abs/2605.26108.

Dengyang Jiang, Dongyang Liu, Zanyi Wang, Qilong Wu, Liuzhuozheng Li, Hengzhuang Li, Xin Jin, Changsheng Lu, Zhen Li, Mengmeng Wang, Steven Hoi, Peng Gao, and Harry Yang. Distribution Matching Distillation Meets Reinforcement Learning. In *European Conference on Computer Vision*, pp. 281–299, 2026. doi: 10.1007/978-3-032-37550-6_16. URL https://doi.org/10.1007/978-3-032-37550-6_16.

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 *Advances in Neural Information Processing Systems*, 2023. doi: 10.52202/075280-1594. URL https://doi.org/10.52202/075280-1594.

Junzhe Li, Yutao Cui, Tao Huang, Weijie Kong, Yiming Cheng, Chuxuan Zeng, Yinping Ma, Chun Fan, Miles Yang, Zhao Zhong, and Liefeng Bo. MixGRPO: Unlocking Flow-based GRPO Efficiency with Mixed ODE-SDE. *arXiv preprint arXiv:2507.21802*, 2025a. URL https://arxiv.org/abs/2507.21802.

Yuming Li, Yikai Wang, Yuying Zhu, Zhongyu Zhao, Ming Lu, Qi She, and Shanghang Zhang. BranchGRPO: Stable and Efficient GRPO with Structured Branching in Diffusion Models. *arXiv preprint arXiv:2509.06040*, 2025b. URL https://arxiv.org/abs/2509.06040.

Zhanhao Liang, Yuhui Yuan, Shuyang Gu, Bohan Chen, Tiankai Hang, Mingxi Cheng, Ji Li, and Liang Zheng. Aesthetic Post-Training Diffusion Models from Generic Preferences with Step-by-step Preference Optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. doi: 10.1109/cvpr52734.2025.01232. URL https://doi.org/10.1109/cvpr52734.2025.01232.

Zhanhao Liang, Tao Yang, Jie Wu, Chengjian Feng, and Liang Zheng. LeapAlign: Post-training Flow Matching Models at Any Generation Step by Building Two-Step Trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23238–23248, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Liang_LeapAlign_Post-training_Flow_Matching_Models_at_Any_Generation_Step_by_CVPR_2026_paper.html.

Xinyao Liao, Wei Wei, Xiaoye Qu, Qiyuan He, Angela Yao, and Yu Cheng. CoCA: Step-level Reward for Free in RL-based T2I Diffusion Model Fine-tuning. *arXiv preprint arXiv:2505.19196*, 2025. URL https://arxiv.org/abs/2505.19196.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow Matching for Generative Modeling. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.02747.

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 *Advances in Neural Information Processing Systems*, 2025. doi: 10.52202/085713-1362. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/3a10c46572628d58cb44fb705f25cbbf-Abstract-Conference.html.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2209.03003.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1711.05101.

Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent Consistency Models: Synthesizing High-Resolution Images with Few-Step Inference. *arXiv preprint arXiv:2310.04378*, 2023. URL https://arxiv.org/abs/2310.04378.

Yihong Luo, Tianyang Hu, Weijian Luo, and Jing Tang. TDM-R1: Reinforcing Few-Step Diffusion Models with Non-Differentiable Reward. *arXiv preprint arXiv:2603.07700*, 2026. URL https://arxiv.org/abs/2603.07700.

Yuhang Ma, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. HPSv3: Towards Wide-Spectrum Human Preference Score. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15086–15095, 2025. doi: 10.1109/ICCV51701.2025.01400. URL https://openaccess.thecvf.com/content/ICCV2025/html/Ma_HPSv3_Towards_Wide-Spectrum_Human_Preference_Score_ICCV_2025_paper.html.

Zichen Miao, Zhengyuan Yang, Kevin Lin, Ze Wang, Zicheng Liu, Lijuan Wang, and Qiang Qiu. Tuning Timestep-Distilled Diffusion Model Using Pairwise Sample Optimization. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/243697ace81f57daef8737ff2c5cffd3-Abstract-Conference.html.

Mihir Prabhudesai, Anirudh Goyal, Deepak Pathak, and Katerina Fragkiadaki. Aligning Text-to-Image Diffusion Models with Reward Backpropagation. *arXiv preprint arXiv:2310.03739*, 2023. URL https://arxiv.org/abs/2310.03739.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. *arXiv preprint arXiv:2205.11487*, 2022. URL https://arxiv.org/abs/2205.11487.

Tim Salimans and Jonathan Ho. Progressive Distillation for Fast Sampling of Diffusion Models. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2202.00512.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial Diffusion Distillation. In *European Conference on Computer Vision*, 2024. doi: 10.1007/978-3-031-73016-0_6. URL https://doi.org/10.1007/978-3-031-73016-0_6.

Yash Savani, Branislav Kveton, Yuchen Liu, Yilin Wang, Jing Shi, Subhojyoti Mukherjee, Nikos Vlassis, and Krishna Kumar Singh. Stepwise Credit Assignment for GRPO on Flow-Matching Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 42007–42017, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Savani_Stepwise_Credit_Assignment_for_GRPO_on_Flow-Matching_Models_CVPR_2026_paper.html.

Christoph Schuhmann. LAION-Aesthetics, 2022. URL https://laion.ai/blog/laion-aesthetics/.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency Models. In *International Conference on Machine Learning*, 2023. URL https://arxiv.org/abs/2303.01469.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion Model Alignment Using Direct Preference Optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. doi: 10.1109/cvpr52733.2024.00786. URL https://doi.org/10.1109/cvpr52733.2024.00786.

Ruoyu Wang, Boye Niu, Xiangxin Zhou, Yushi Huang, Tongliang Liu, and Chi Zhang. Exploring the Design Space of Reward Backpropagation for Flow Matching. *arXiv preprint arXiv:2606.11075*, 2026. URL https://arxiv.org/abs/2606.11075.

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. URL https://arxiv.org/abs/2503.05236.

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. URL https://arxiv.org/abs/2306.09341.

Xiaoshi Wu, Yiming Hao, Manyuan Zhang, Keqiang Sun, Zhaoyang Huang, Guanglu Song, Yu Liu, and Hongsheng Li. Deep Reward Supervisions for Tuning Text-to-Image Diffusion Models. In *European Conference on Computer Vision*, 2024. doi: 10.1007/978-3-031-73010-8_7. URL https://doi.org/10.1007/978-3-031-73010-8_7.

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 *Advances in Neural Information Processing Systems*, 2023. doi: 10.52202/075280-0700. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/33646ef0ed554145eab65f6250fab0c9-Abstract-Conference.html.

Shuchen Xue, Chongjian Ge, Shilong Zhang, Yichen Li, and Zhi-Ming Ma. Advantage Weighted Matching: Aligning RL with Pretraining in Diffusion Models. *arXiv preprint arXiv:2509.25050*, 2025a. URL https://arxiv.org/abs/2509.25050.

Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, and Ping Luo. DanceGRPO: Unleashing GRPO on Visual Generation. *arXiv preprint arXiv:2505.07818*, 2025b. URL https://arxiv.org/abs/2505.07818.

Kai Yang, Jian Tao, Jiafei Lyu, Chunjiang Ge, Jiaxin Chen, Weihan Shen, Xiaolong Zhu, and Xiu Li. Using Human Feedback to Fine-tune Diffusion Models without Any Reward Model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8941–8951, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Yang_Using_Human_Feedback_to_Fine-tune_Diffusion_Models_without_Any_Reward_CVPR_2024_paper.html.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T. Freeman. Improved Distribution Matching Distillation for Fast Image Synthesis. In *Advances in Neural Information Processing Systems*, 2024a. doi: 10.52202/079017-1505. URL https://doi.org/10.52202/079017-1505.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T. Freeman, and Taesung Park. One-step Diffusion with Distribution Matching Distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024b. doi: 10.1109/cvpr52733.2024.00632. URL https://doi.org/10.1109/cvpr52733.2024.00632.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14483–14494, 2025. doi: 10.1109/CVPR52734.2025.01350. URL https://openaccess.thecvf.com/content/CVPR2025/html/You_Teaching_Large_Language_Models_to_Regress_Accurate_Image_Quality_Scores_CVPR_2025_paper.html.

Ziyi Zhang, Li Shen, Sen Zhang, Deheng Ye, Yong Luo, Miaojing Shi, Dongjing Shan, Bo Du, and Dacheng Tao. Aligning Few-Step Diffusion Models with Dense Reward Difference Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(7):7375–7386, 2026. doi: 10.1109/tpami.2026.3665753. URL https://doi.org/10.1109/tpami.2026.3665753.

Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su, Stefano Ermon, Jun Zhu, and Ming-Yu Liu. DiffusionNFT: Online Diffusion Reinforcement with Forward Process. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/d8e68ddfe22520b45b8fb8d5cbde5a21-Abstract-Conference.html.

Yujie Zhou, Pengyang Ling, Jiazi Bu, Yibin Wang, Yuhang Zang, Jiaqi Wang, Li Niu, and Guangtao Zhai. Fine-Grained GRPO for Precise Preference Alignment in Flow Models. *arXiv preprint arXiv:2510.01982*, 2025. URL https://arxiv.org/abs/2510.01982.

A Implementation and Protocol Details

A.1 Student and LoRA Configuration

We use FLUX.1-dev (Black Forest Labs, 2024) as the backbone and expand its output projection into the K=8-component GM head defined in Section 3.1. The original projection weights and biases are replicated to initialize the component means. Mixture logits are initialized uniformly, and the log-scale bias is set to $\log 0.05$. The replicated mean biases are perturbed with 0.05-scale noise to distinguish the components at initialization. The GM output projection and all backbone weights remain frozen; only LoRA parameters are updated. The initial validation reward is 0.60 for the base GM conversion and 12.20 for the official distilled student. LoRA uses rank 32, α=32, and dropout 0.05, and is attached to the modules `proj_mlp`, `proj_out`, `ff.net.0.proj`, `ff.net.2`, `ff_context.net.0.proj`, `ff_context.net.2`, `timestep_embedder.linear_1`, and `timestep_embedder.linear_2`.

A.2 GM Parameterization

One transformer evaluation predicts component velocities $u_{i,k}$, log-scales $\ell_{i,k}$, and logits $b_{i,k}$. With $\sigma_i = \sigma_i^{\text{src}}$, these define

$$\begin{aligned} A_{i,k} &= \operatorname{softmax}_k(b_{i,k}), \qquad \mu^0_{i,k} = x_i - \sigma_i u_{i,k}, \quad v^0_{i,k} = \sigma_i^2 \exp(2\ell_{i,k}), \\ q_{\theta,i}(z \mid x_i) &= \textstyle\sum_{k=1}^K A_{i,k} \mathcal{N}(z; \mu^0_{i,k}, v^0_{i,k} I). \end{aligned} \tag{11}$$

For clarity, the notation omits the factorization over latent locations used in the implementation. The posterior update in equation 1 reuses this mixture throughout the interval.

A.3 Deployment Sampler

Four-step training trajectories and evaluations use the same sampler. We use the flow-map SDE scheduler from the π-Flow codebase (Chen et al., 2026) with N=4 outer steps, scheduler shift 3.2, and final-step scale 0.5. Each interval is integrated with the analytic velocity field. The total budget of 130 substeps is allocated in proportion to interval length, giving 37/37/37/19 for the four intervals. The SDE temperature is 0.3 on the first three intervals and 0 on the last. The guidance scale is 3.5 and the resolution is 512×512. The default evaluation sampler seed is 1223627; GenEval uses seed 42, as specified in Appendix A.6.

A.4 Optimization

We train with AdamW (Loshchilov & Hutter, 2019) at learning rate 10^{-4}, (β_1, β_2)=$(0.9, 0.95)$, 100 warmup steps, and gradient clipping at 1.0. We train for 500 optimizer steps with 64 trajectories per step and evaluate the resulting checkpoint. Training uses bf16 precision and fully sharded data parallelism (FSDP) on 8 GPUs with 80 GB of memory each.

A.5 Checkpoints and Selection

Checkpoints are saved every 50 steps. Validation reward is also measured every 50 steps on held-out prompts using the deployment sampler. We evaluate STEP, all five component ablations, DRaFT, LeapAlign+substep, and Flow-GRPO at optimizer step 500. These methods therefore use the same training budget of 500×64 trajectories, although the computation per update differs. The zero-RL and two-stage references use the checkpoints from the original two-stage experiments. They have the same inference budget of four NFEs, with training schedules specific to each procedure. All four qualitative variants in Figure 4 use step 500, matching Table 3. Other qualitative comparisons and gradient diagnostics retain their original evaluation provenance. Timing statistics use earlier training logs. No test metric is used for checkpoint selection.

A.6 Seeds

Multi-seed runs use training seeds 42, 43, and 44. The LeapAlign step-selection procedure uses seeds 114, 115, and 116 for the three runs. GenEval generation uses seed 42 for every model.

A.7 Scorers

All scorers are frozen and shared across models. Text rendering is scored by the OCR reward of Liu et al. (2025) with PaddleOCR 3.7.0 (Cui et al., 2025) on 1018 prompts. GenEval (Ghosh et al., 2023) uses its official detection pipeline on 553 prompts with 4 images per prompt. Additional evaluation metrics are HPSv2.1 (Wu et al., 2023), PickScore (Kirstain et al., 2023), ImageReward (Xu et al., 2023), and the alignment and image-quality scores of UnifiedReward (Wang et al., 2025). PickScore uses logit_scale$\times$cos similarity. Table 1 reports the raw values; Table 6 divides them by 26, following Wang et al. (2026). We assess image quality with the LAION aesthetic predictor (Schuhmann, 2022) and DeQA-Score (You et al., 2025), abbreviated as DeQA in the tables.

A.8 Baseline Implementations

The direct-gradient baselines and ablations (DRaFT, LeapAlign with the substep reward, LeapProbe, Head-ST, and ChainB) share the backbone, prompt distribution, reward model, and deployment sampler. LeapAlign+substep denotes our adaptation to the GM sampler, as distinct from the published model evaluated under its original protocol. The component ablations modify the gradient construction or remove a loss term from Section 3. Sharing a reward model does not imply an identical objective: STEP includes two auxiliary reward evaluations and teacher regularization, whereas DRaFT and Flow-GRPO use one reward evaluation and no teacher evaluations (Table 8). Flow-GRPO (Liu et al., 2025) is adapted to the same backbone, data, reward, and four-NFE sampler, with groups of 8 trajectories per prompt. The original method uses 10 sampling steps during training and 40 at inference. Our baseline and component ablation evaluations all use optimizer step 500.

B Full Results

This appendix provides additional quantitative results for Section 4. Published results under other protocols are reported separately. STEP, its component ablations, and the adapted baselines are evaluated at optimizer step 500. The zero-RL student and two-stage procedures use the reference checkpoints in Appendix A.5.

B.1 Multi-Seed OCR Summary

Table 4 reports the OCR scores from the main comparison alongside means across training seeds. STEP and the variant without teacher regularization both average 0.550, compared with 0.530 for LeapAlign+substep. These means describe performance across the training seeds in Appendix A.6; differences between means alone do not establish statistical significance.

Table 4: Main OCR and multi-seed mean OCR at optimizer step 500.

Method	Main OCR	Multi-seed mean
STEP	0.580	0.550
– teacher anchor	0.600	0.550
LeapAlign+substep	0.560	0.530

B.2 GenEval Results by Task for Two-Stage Methods

Table 5 reports per-task GenEval scores for the zero-RL student and two-stage methods. Evaluation uses 553 prompts, 4 images per prompt, and generation seed 42. The main comparison and component ablations appear in Tables 1 and 3.

For distill-then-RL, the full objective, auxiliary-only variant, and variant without auxiliary rewards or teacher regularization achieve validation rewards of 15.49, 15.74, and 14.45, respectively. The zero-RL student scores 12.20 on that validation set. Validation reward uses a separate prompt set from the held-out HPSv3 evaluation.

Table 5: Per-task GenEval accuracy (%) for the distilled reference and two-stage training routes. These rows use the reference checkpoints described in Appendix A.5.

Method	single	two	counting	colors	position	color attr	overall
π-Flow (zero RL)	71.88	47.47	38.75	49.47	14.25	15.50	39.55
RL at 16 NFEs, deployed at 4	81.88	50.51	44.38	54.52	12.50	17.50	43.55
RL-then-distillation	89.38	56.06	41.25	56.91	13.75	21.50	46.48
distill-then-RL (full)	79.06	48.74	37.81	53.19	11.50	17.75	41.34
distill-then-RL (aux only)	83.75	46.97	32.50	44.95	15.00	11.50	39.11
distill-then-RL (no aux, no anchor)	77.81	50.51	35.94	53.19	16.50	14.50	41.41

B.3 Evaluation with Multiple Reward Models

Table 6 extends Table 1 to all methods evaluated on the held-out set (400 HPDv2 test prompts $\times 4$ images). All reward models are frozen. HPSv3 is used during training; the remaining reward models are used only for evaluation. All images are generated with the four-NFE, 512×512 deployment sampler in Appendix A.3. PickScore is divided by 26, following Wang et al. (2026); Table 1 reports the raw logits. Normalized values for the main baselines are computed from the raw values at the reported precision.

Table 6: Reward scores on held-out HPDv2 prompts. HPSv3† is used for training; all other metrics are used only for evaluation. All models use our four-NFE, 512×512 sampler. PS denotes PickScore divided by 26. Checkpoints follow Appendix A.5.

Method	HPSv3†	HPSv2.1	PS	IR	UR-A	UR-IQ
π-Flow (zero RL)	13.47	0.300	0.865	0.988	0.674	0.749
DRaFT	14.20	0.330	0.873	1.250	0.690	0.730
Flow-GRPO	15.10	0.350	0.888	1.360	0.690	0.750
LeapAlign+substep	15.50	0.360	0.877	1.280	0.700	0.750
LeapProbe	15.30	0.350	0.870	1.280	0.680	0.730
Head-ST	15.35	0.355	0.890	1.320	0.700	0.750
STEP − anchor	15.50	0.360	0.870	1.380	0.700	0.760
distill-then-RL (full)	15.02	0.330	0.869	1.104	0.681	0.766
distill-then-RL (aux only)	14.95	0.335	0.857	1.178	0.664	0.734
STEP (ours)	15.80	0.380	0.885	1.350	0.720	0.750

These results are consistent with the component ablations on GenEval and OCR. Removing the chain or using zero-depth probes lowers HPSv3 from 15.80 to 15.30 or 15.35, respectively. Full STEP also has the highest HPSv2.1 and UR-Align scores in this comparison. Method rankings depend on the metric: Head-ST has higher normalized PickScore, and removing the teacher regularizer increases ImageReward and UR-IQ. These comparisons show that the full objective and its variants favor different evaluation criteria. They do not identify a single variant that dominates across all scorers.

B.4 Image Quality

Table 7 reports Aesthetic and DeQA scores on HPDv2 and DrawBench. STEP exceeds the zero-RL student on both metrics and both prompt sets. On HPDv2, removing the teacher regularizer leaves Aesthetic unchanged and increases DeQA from 4.30 to 4.40. The variants initialized from the distilled student also achieve competitive DeQA scores.

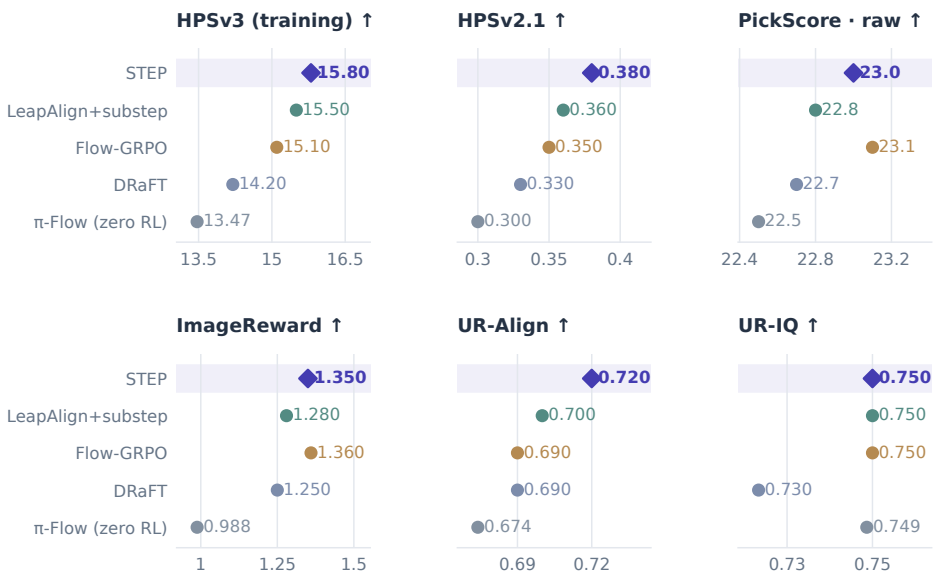

Figure 5: Reward scores for the main four-step comparison. Each panel reports the original score scale; PickScore uses raw logits. STEP and adapted baselines use optimizer step 500. The official zero-RL student provides the distilled reference.

Table 7: Aesthetic and DeQA scores on the held-out HPDv2 set and on DrawBench prompts (200 prompts ×4 images). All models use our four-NFE, 512×512 sampler.

	HPDv2 test ×4		DrawBench ×4	
Method	Aes	DeQA	Aes	DeQA
π-Flow (zero RL)	6.12	4.25	5.61	4.33
STEP (ours)	6.40	4.30	6.00	4.40
STEP − anchor	6.40	4.40	–	–
distill-then-RL (full)	6.34	4.37	–	–
distill-then-RL (aux only)	–	4.48	–	–

B.5 Cost Details

Table 8 details the computation required by each method. Table 9 reports mean time per optimizer step over the training-log intervals specified in its caption. The timing logs come from earlier training runs, separate from the 500-step benchmark runs. The benchmark comparison matches trajectory counts, not GPU time.

Table 8: Training computation with 64 trajectories per optimizer step. Transformer counts are per trajectory and list, in order, student evaluations without gradients, student evaluations with gradients, and frozen teacher evaluations. Each scored image requires a decoder and reward evaluation.

Quantity	STEP	DRaFT	Flow-GRPO
Transformer evaluations	4/4/4	0/4/0	8/8/0
Reward evaluations per trajectory	1 (+2 aux)	1	1
Trajectories per prompt	1	1	8
Scored images per step	$\leq$ 192	64	64

Table 9: Mean time per optimizer step from earlier training runs. The DRaFT log covers steps 0–983 and the Flow-GRPO log covers steps 0–503. All other averages use the complete training run.

Method	Time per step (s)
STEP (ours)	50.3
DRaFT	39.2
Flow-GRPO	19.5
distill-then-RL (full objective)	48.6
distill-then-RL (aux only)	∼41
distill-then-RL (no aux, no anchor)	37.0

B.6 Published Results under External Protocols

The following tables provide published results for context. They use the original prompt sets, training rewards, and sampling settings, so they are not directly comparable with Table 1. Results for different training tasks correspond to separately trained models, including those reported for the same method.

Table 10: Published verifiable alignment results. Flow-GRPO uses SD3.5-M with 40 inference steps; its GenEval and OCR results come from separate task-specific models (Liu et al., 2025, Table 2). LeapAlign uses FLUX.1-dev at 720^2 and 50 inference steps (Liang et al., 2026, Table 2).

Method	Training task	NFE	GenEval	OCR
Flow-GRPO (with KL)	Composition	40	0.95	–
Flow-GRPO (with KL)	Text rendering	40	–	0.92
LeapAlign	Composition	50	0.742	–

Table 11: Published preference and quality scores by training task. Sources are the same as Table 10. PS uses raw PickScore logits. Flow-GRPO rows use KL regularization and report quality and preference scores on DrawBench; LeapAlign reports preference scores on HPDv2. LeapAlign's training reward is HPSv2.1; HPSv3 is held out for that method. Dashes indicate unavailable entries.

Method	Task	HPSv3	HPSv2.1	PS	IR	Aes	DeQA
Flow-GRPO	Composition	–	–	22.37	1.03	5.25	4.01
Flow-GRPO	Text	–	–	22.44	0.95	5.32	4.06
LeapAlign	Preference	15.77	0.409	23.71	1.51	–	–

C Additional Analysis and Discussion

C.1 Fusion Coefficient Bound

The distances in equation 3 satisfy $d_i \geq \tau > 0$. Summing the nonnegative coefficients of equation 4 gives

$$\sum_{i=0}^{N-1} c_i = \frac{\sum_i d_i^{-1}}{(\sum_j d_j)(\sum_k d_k^{-1})} = \frac{1}{\sum_j d_j} \leq \frac{1}{N\tau}.$$

For $N = 4$ and $\tau = 0.1$, the sum of the coefficients is at most 2.5. Each coefficient also lies between 0 and 2.5. The bound applies to the fusion weights. Bounding the full parameter gradient would additionally require control of the probe and reward Jacobians in equation 6. The factorization $c_i = \gamma w_i$ in Section 3.2 also separates relative allocation from overall scale. The normalized weights satisfy $\sum_i w_i = 1$, whereas $\gamma = 1/\sum_i d_i$ varies across samples. Uniform weights with the same total coefficient would set $c_i = \gamma/N$. Even at equal total coefficient, different probe Jacobians can yield different parameter gradient norms. The coefficient bound alone establishes neither an accurate gradient direction nor convergence of the training procedure.

C.2 Hinge Margin Saturation

The terminal loss in equation 7 contributes a reward gradient only for samples below the margin M=11. For samples above the margin, the main loss and its gradient are zero. Thus, a differentiable path from the terminal reward to each probe does not imply that every training sample updates all heads through this term. Late in training, the batch mean reward exceeds 15. The mean alone does not determine the fraction of saturated samples or the magnitude of the batch gradient.

The three objective terms have different activation conditions. Unsaturated terminal samples contribute through the fused probes. Auxiliary rewards supervise intermediate estimates from one head per update and can remain below the margin even when the terminal reward exceeds it. Teacher regularization is independent of the reward margin. The decline in OCR and GenEval when auxiliary rewards are removed (Table 3) establishes their value within the combined objective. It does not quantify the share of improvement attributable to terminal supervision.

The same hinge formulation (Liang et al., 2026) also makes initialization relevant to objective activity. The base GM conversion and distilled student have initial mean validation rewards of 0.60 and 12.20, respectively (Appendix A.1). Neither mean determines the fraction of training samples below the margin. Logs from the two-stage runs without teacher regularization include updates with zero loss and zero gradient norm. These observations motivate considering loss saturation when interpreting the training routes; they do not isolate its effect from initialization or training history.

Saturation also affects the gradient diagnostics. We measure the gradient norms in Figure 6 using the unsaturated terminal loss $-g\,\mathrm{mean}(r)$. The diagnostic evaluates the gradient paths without masking samples above the hinge margin. For an unsaturated sample, its reward gradient agrees with that of the training loss. For a batch containing saturated samples, the two gradient sums generally differ. Figure 6 therefore characterizes the surrogate's connectivity, not the activity of the three objective terms during training.

C.3 Gradient Diagnostics by Head and Prompt

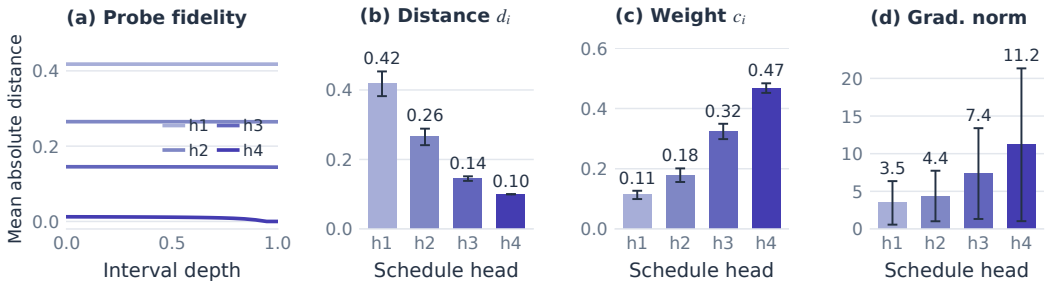

Figure 6: **Gradient diagnostics on 16 held-out prompts.** Measurements use the earlier diagnostic checkpoint. (a) Mean distance from each probe to the terminal latent as a function of depth within the interval. (b,c) Mean clipped probe distances d_i and fusion coefficients c_i; error bars show standard deviations across prompts. (d) Per-head LoRA gradient norms under the unsaturated terminal loss $-g\,\mathrm{mean}(r)$, with mean and standard deviation. The unsaturated loss measures terminal gradient propagation without hinge saturation; Appendix C.2 discusses the training objective.

We evaluate the gradient construction on 16 held-out HPDv2 prompts using the earlier diagnostic checkpoint. These measurements are separate from the 500-step benchmark evaluation. Figures and tables label heads from 1 to N; head h corresponds to the method index $i = h - 1$. The heads share the same LoRA parameters. Per-head norms describe gradient paths through different schedule positions, not disjoint parameter groups, and cannot be added to obtain the norm of the full parameter update. Table 12 reports results for each prompt and summaries by head. Gradient norms strictly increase along the chain for 15 of 16 prompts. The remaining prompt has a single inversion between the first two heads. The final head reaches the distance lower bound (d_3=0.100 for every prompt), so differences below this threshold do not affect its weight. Probe-to-terminal distances are nearly constant within the first three intervals: the GM posterior produces similar $\widehat{x}_0$ estimates

at different depths. The distance approaches zero at the end of the final interval. Similar latent distances do not imply similar probe Jacobians. These measurements characterize probe values and gradient propagation; they do not compare the effectiveness of random and fixed interior depths. The diagnostic uses bf16 weights; spot checks with fp32 agree on d_i to within 0.001. Every prompt satisfies $d_0 > d_1 > d_2 > d_3$ and, consequently, $c_0 < c_1 < c_2 < c_3$.

Table 12: Gradient diagnostics for STEP. Top: results for each held-out prompt, including probe distances d_i, fusion coefficients c_i, and per-head LoRA gradient norms under the unsaturated terminal loss. The distance for head 4 equals the lower bound τ=0.1 for every prompt. The row marked * has a single inversion in gradient norm between the first two heads. Bottom: mean ± standard deviation across the 16 prompts; n_i is the substep count for interval i.

	d_i				c_i				grad norm			
Prompt	1	2	3	4	1	2	3	4	1	2	3	4
Dwayne Johnson wrestles Jesus in WWE	0.385	0.240	0.141	0.100	0.126	0.202	0.344	0.484	2.77	4.15	6.61	9.84
Anime man in flight uniform	0.409	0.259	0.146	0.100	0.115	0.182	0.324	0.472	0.39	0.41	0.68	0.93
Wojak over a sea of memes	0.422	0.259	0.148	0.100	0.111	0.181	0.317	0.468	0.86	1.12	1.78	2.93
Ralsei and Asriel eating pizza	0.466	0.290	0.150	0.100	0.096	0.154	0.298	0.446	1.14	2.30	2.95	4.88
Anime mecha robot portrait	0.478	0.293	0.148	0.100	0.092	0.150	0.298	0.441	0.69	1.07	1.14	1.61
Chloe Grace by Jean Giraud	0.345	0.228	0.131	0.100	0.145	0.218	0.382	0.499	0.17	0.26	0.37	0.42
Raccoon riding an oversized fox	0.428	0.285	0.154	0.100	0.108	0.163	0.301	0.463	5.61	6.71	11.61	16.16
Chucky doll as Beetlejuice	0.428	0.264	0.144	0.100	0.108	0.175	0.322	0.463	0.13	0.21	0.32	0.47
2B eating a bagel	0.463	0.292	0.152	0.100	0.097	0.153	0.294	0.448	8.60	9.31	15.06	26.41
Groot depicted as a flower	0.425	0.274	0.146	0.100	0.109	0.169	0.316	0.463	7.42	9.46	19.89	37.68
Two women with purple hair	0.392	0.223	0.135	0.100	0.123	0.216	0.356	0.481	4.04	6.33	9.20	13.71
Girl with pink pigtails	0.400	0.266	0.144	0.100	0.118	0.178	0.329	0.473	4.51	6.42	10.70	14.74
Cat in a tutu dancing*	0.419	0.274	0.147	0.100	0.111	0.170	0.317	0.466	7.77	7.34	11.64	16.18
Wicked witch casting fireball	0.400	0.263	0.143	0.100	0.119	0.180	0.331	0.474	4.80	5.64	10.32	13.21
Link fights an octorok	0.457	0.299	0.156	0.100	0.099	0.150	0.289	0.450	0.67	0.82	1.71	2.84
Cat with two horns	0.367	0.229	0.136	0.100	0.134	0.215	0.362	0.492	5.87	8.54	13.75	16.92

Head	n_i	d_i	c_i	grad norm
1 (highest noise)	37	0.418 ± 0.036	0.113 ± 0.014	3.47 ± 2.89
2	37	0.265 ± 0.024	0.179 ± 0.023	4.38 ± 3.35
3	37	0.145 ± 0.007	0.324 ± 0.025	7.36 ± 6.03
4 (lowest noise)	19	0.100 ± 0.000	0.468 ± 0.016	11.18 ± 10.14

D Extended Qualitative Results

This appendix presents additional examples from HPDv2, OCR, GenEval, and DrawBench to complement the aggregate results in Appendix B. Evaluation provenance is described in Appendix A.5. The examples use distinct prompts and images, with no repetition of Figures 3 and 4.

Table 13: Full prompts for Figure 3. Gen. is the zero-based generation index, shared by all methods within a column.

Example	Gen.	Full prompt
(a)	1	A portrait of an anime mecha robot with a Japanese town background and a starred night sky.
(b)	1	A raccoon riding an oversized fox through a forest in a furry art anime still.
(c)	0	A neon bike rental sign glowing ”Ride the City” on a wet night street.
(d)	0	A product package prominently labeled ”Eco Friendly Materials Used”.
(e)	0	a photo of four vases
(f)	0	a photo of four hot dogs

Table 14: Full prompts for the main-text qualitative ablations in Figure 4. All four variants use checkpoint 500, matching the quantitative ablations in Table 3.

Example	Full prompt
(a)	A portrait of two women with purple hair flying in different directions against a dark background.
(b)	Wicked witch casting fireball dressed in green with screaming expression.

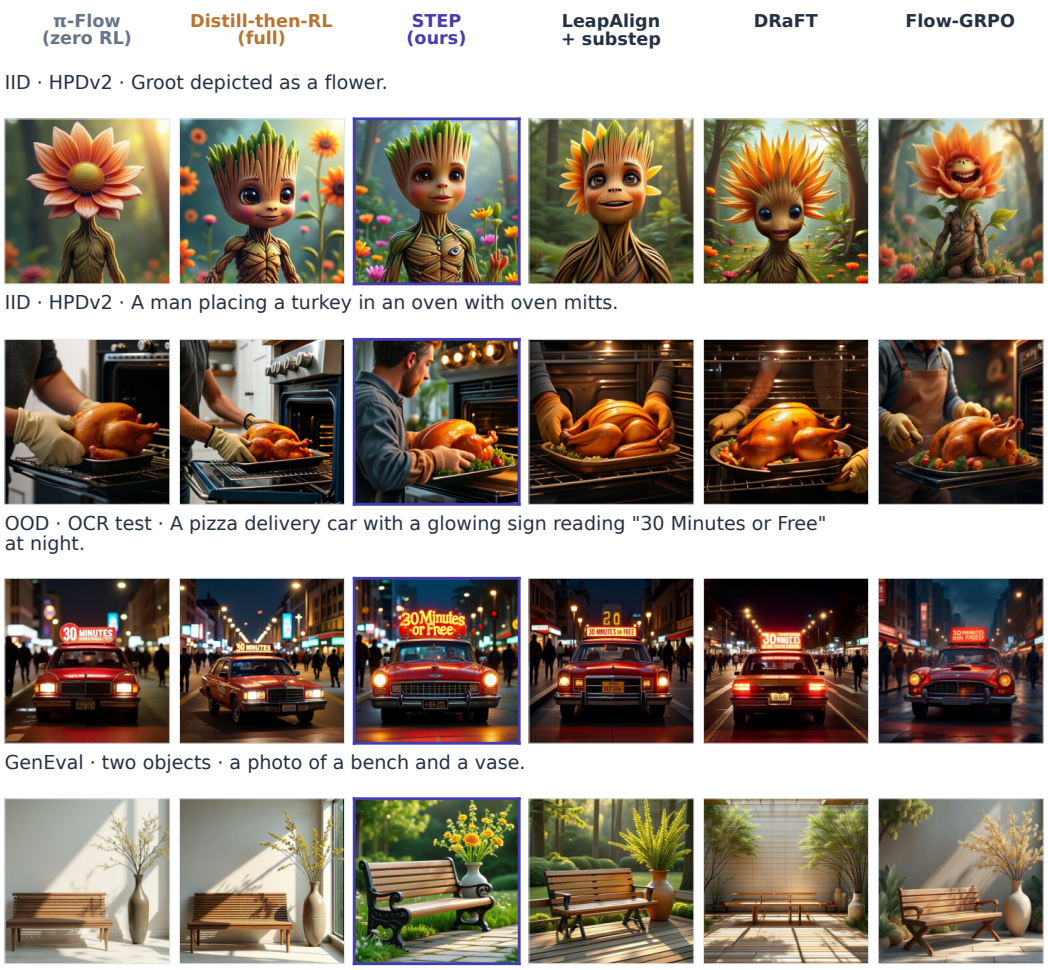

Figure 7: Additional examples from HPDv2, OCR, and GenEval. Each row compares the same prompt across six methods.

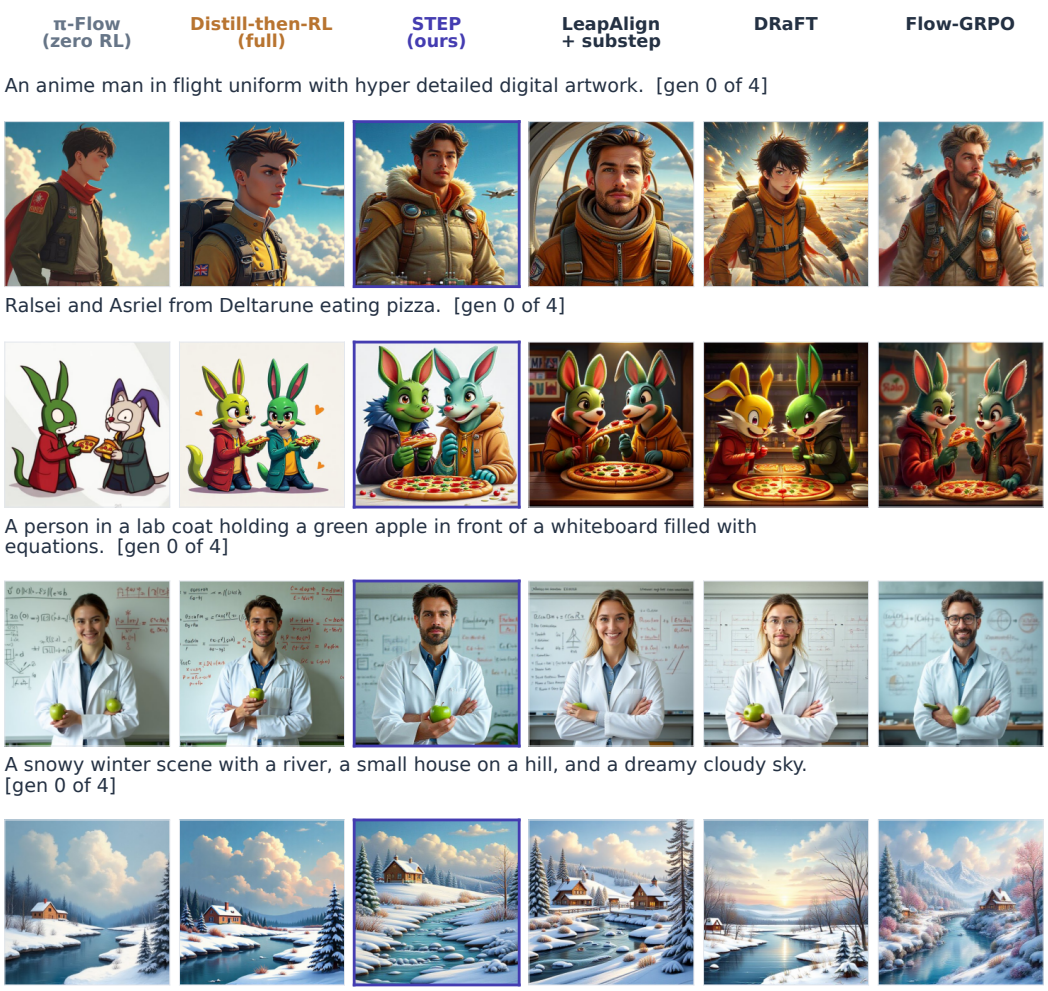

Figure 8: Four additional HPDv2 test prompts. Each row uses generation index 0 across models. Examples were selected for agreement between the generated images and their prompts.

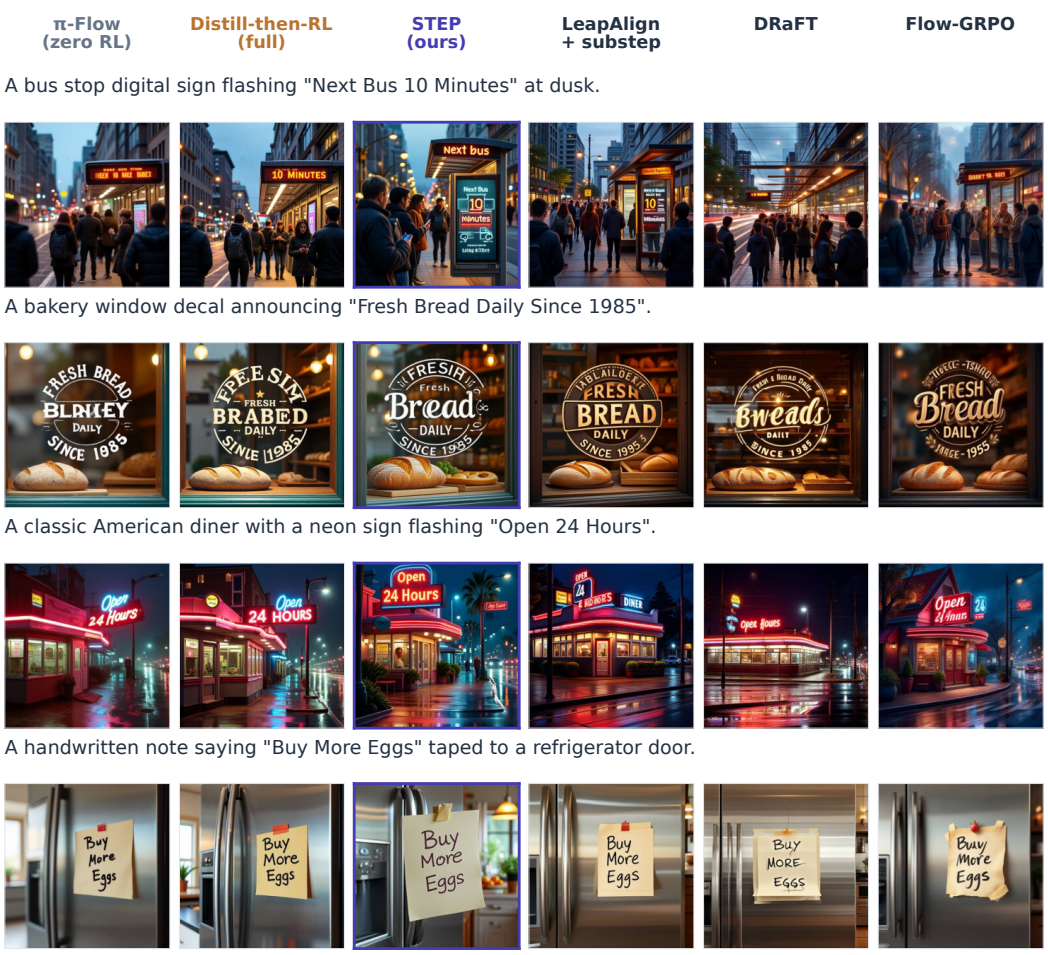

Figure 9: Four additional OCR test prompts with the same methods as Figure 3. The examples illustrate differences in rendering the requested text. All methods correctly render the short string in the final row.

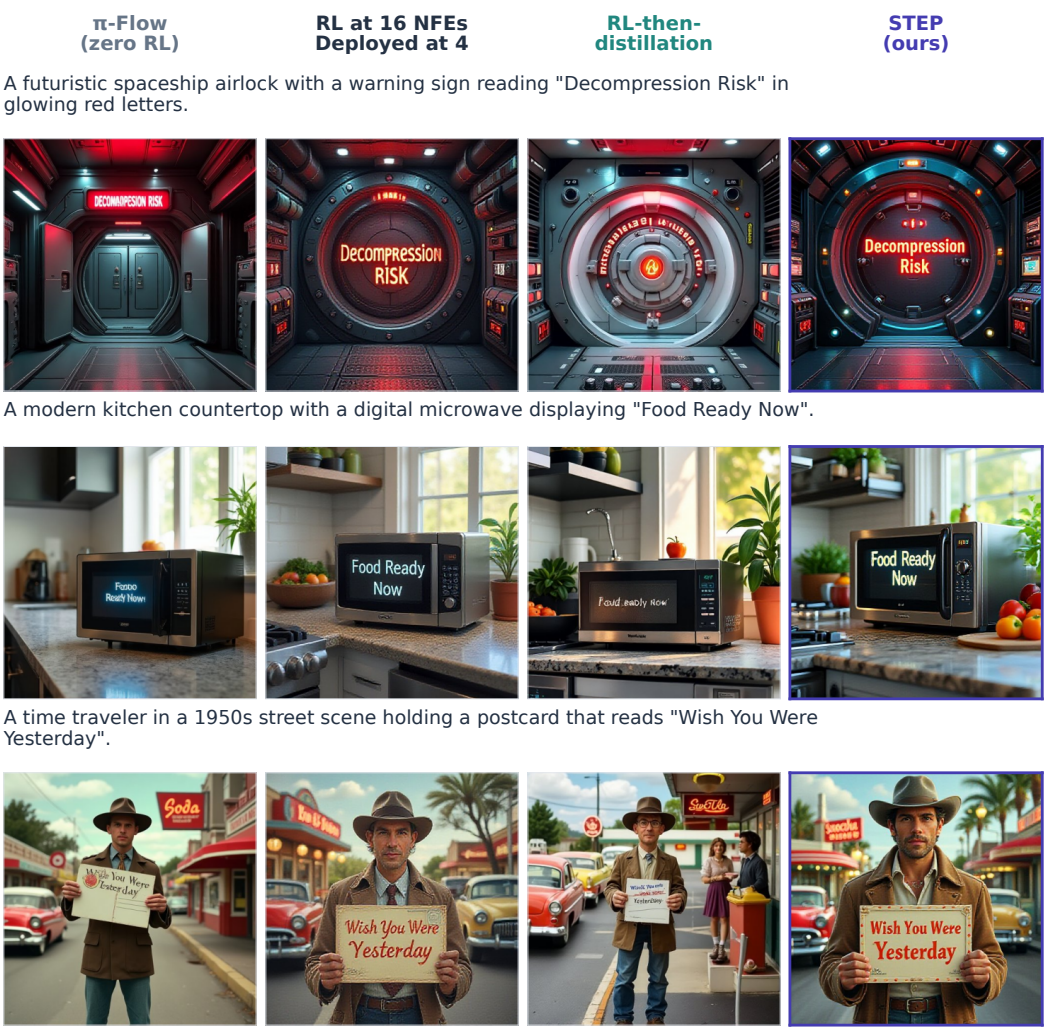

Figure 10: Comparison of training procedures on three OCR prompts. The second column uses the model trained at 16 NFEs but deployed at four NFEs. The third column uses its distilled student. All four models use four NFEs to generate the displayed images.

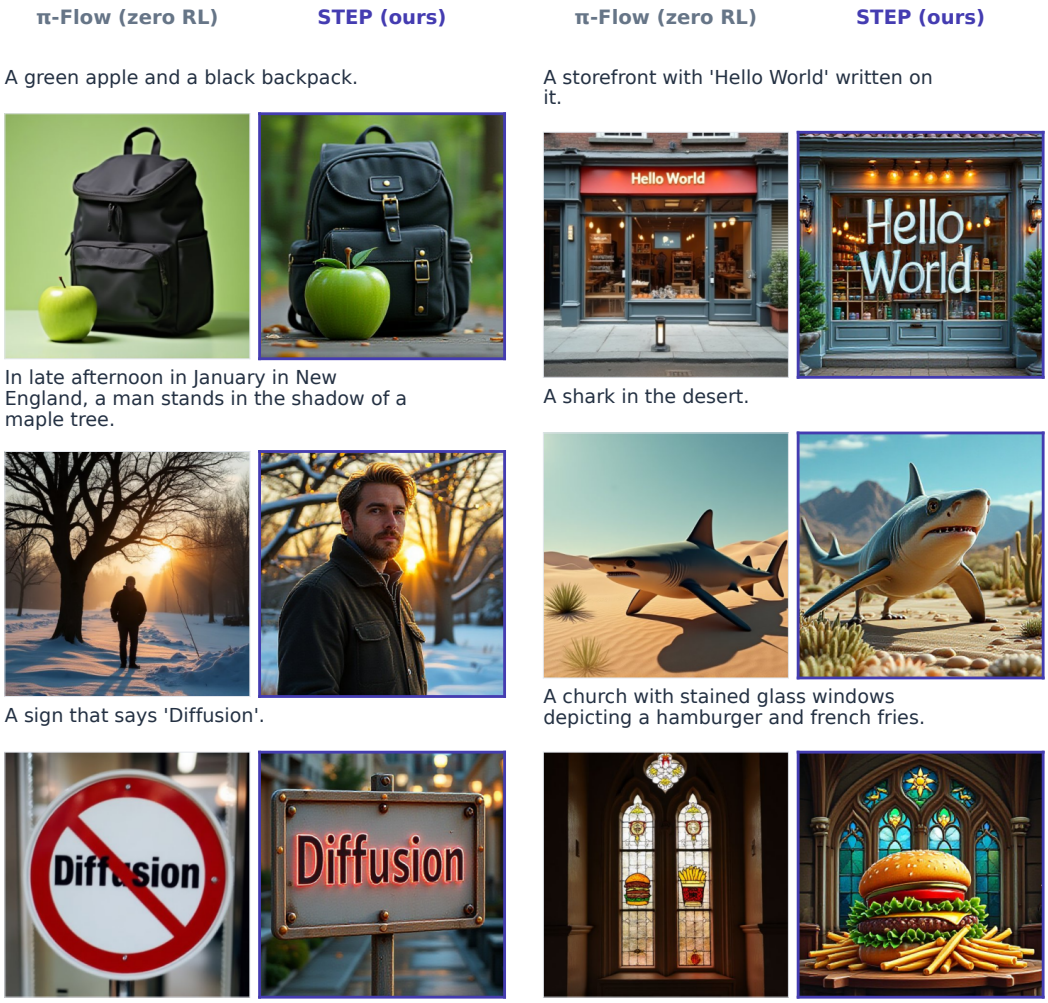

Figure 11: DrawBench comparisons between the official distilled student and STEP on six prompts. Aggregate Aesthetic and DeQA results appear in Table 7.

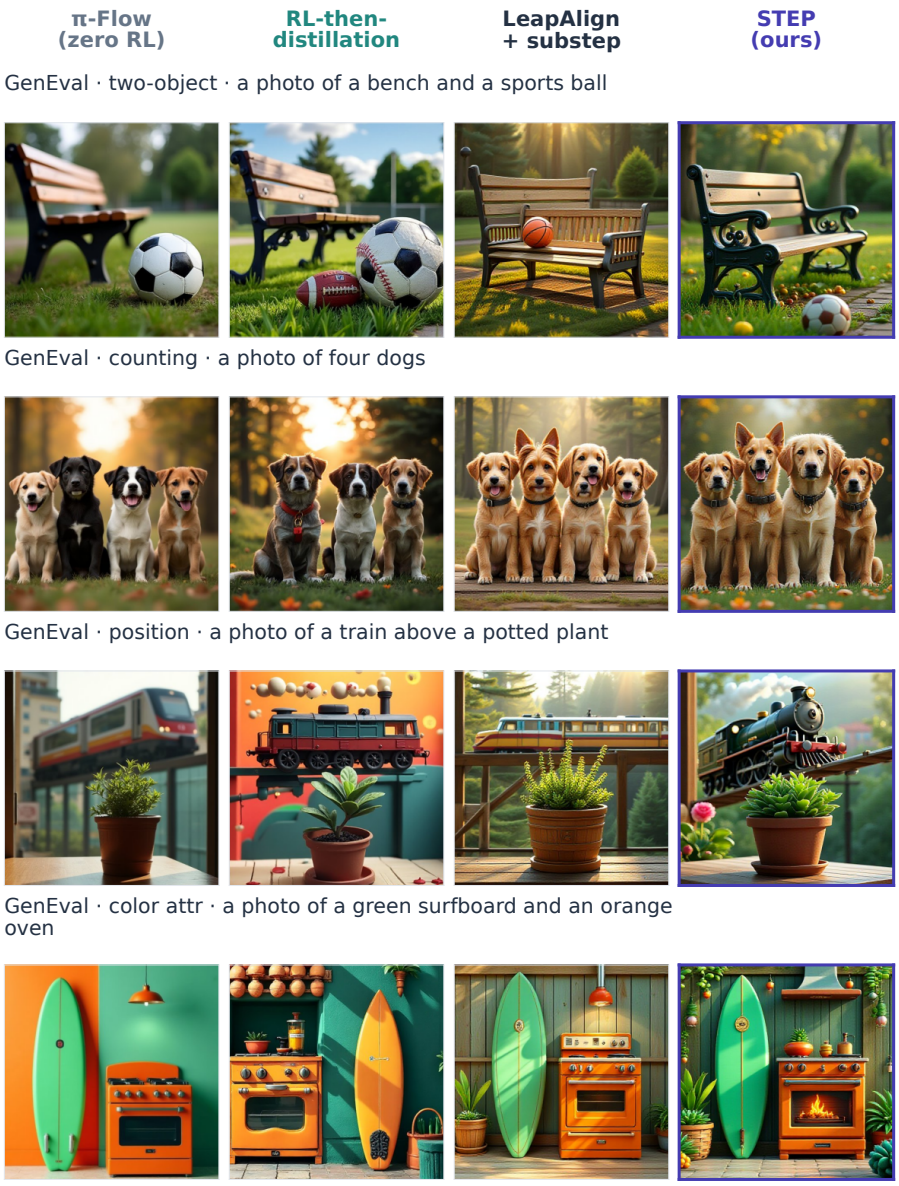

Figure 12: GenEval examples selected from all 553 prompts using the automatic evaluator. For each selected prompt, all four STEP images satisfy the evaluation criteria, while at least one image from each of π-Flow and RL-then-distillation fails. One image is shown per model, so a displayed baseline image may be correct even when another image for the same prompt fails. The examples cover two-object generation, counting, position, and color-attribute binding.

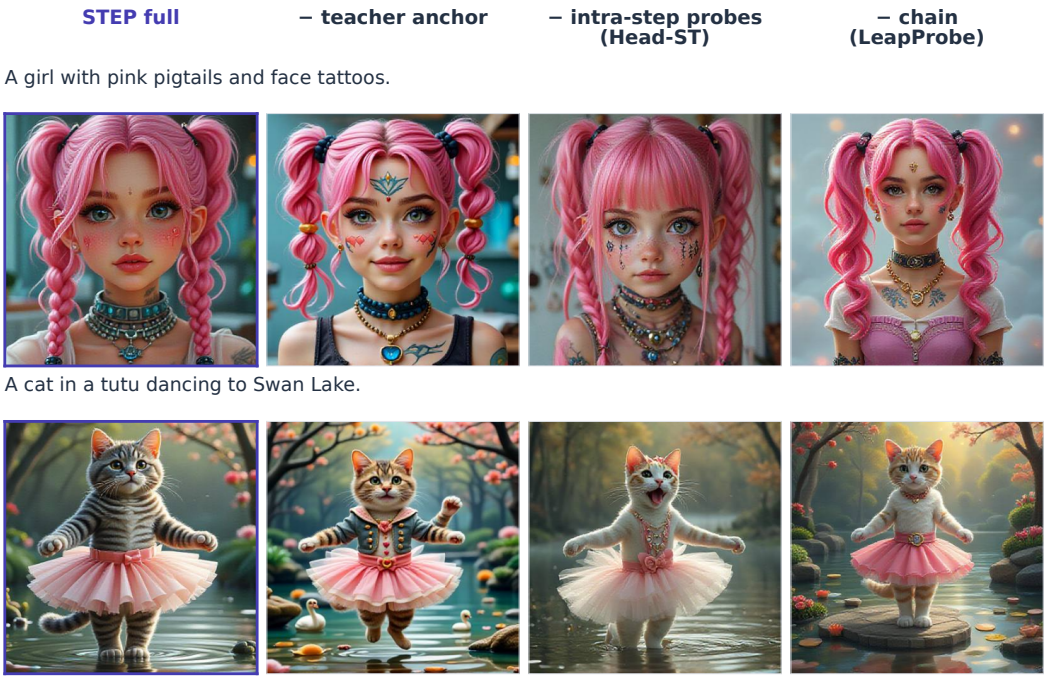

Figure 13: Additional component ablations on two HPDv2 prompts not shown in Figure 4. The variants remove the teacher anchor, replace the intra-step probes with zero-depth probes (Head-ST), or remove the chain (LeapProbe). Their aggregate scores are in Table 3.