LATENT REWARD REGISTERS FOR DIFFUSION PREFERENCE ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Aligning diffusion models with human preferences usually relies on a sparse terminal reward evaluated on the final generated samples, which creates a severe temporal credit-assignment problem across the denoising process. We propose *Latent Reward Registers*, a mechanism that estimates terminal preference directly from intermediate noisy latents. Learnable, position-free register tokens are prepended as an auxiliary read path to a frozen Diffusion Transformer (DiT), extracting preference signals without altering the generator's hidden states or velocity field. The resulting dense, differentiable reward field spans the full denoising trajectory and supports two alignment strategies. For training, *Reward-Gradient On-Policy Distillation* (RG-OPD) converts this dense reward field into per-step targets at states visited by the current generator, replacing rollout-intensive policy gradients with direct on-policy distillation. For inference, *Reward-Guided Sampling* (RGS) steers trajectories with magnitude-matched reward-gradient corrections and no parameter updates. Empirically, at high noise levels (t=0.8) the registers reach the highest pairwise accuracy among the evaluated latent reward models. RG-OPD outperforms online reinforcement learning baselines while reducing GPU hours by up to 33×. RGS achieves significant reward improvement with a favorable reward–quality balance against training-free baselines.

1 INTRODUCTION

Aligning diffusion and flow-matching models with human preferences is most commonly formulated as reward maximization: an external reward model assigns a scalar score to the final generated image, and the generator is trained to maximize this score (Lee et al., 2023; Clark et al., 2024; Prabhudesai et al., 2023; Black et al., 2023; Liu et al., 2025; Zheng et al., 2026). Regardless of the specific reward optimization method, they share a fundamental bottleneck: the reward is evaluated only on the fully generated image at the end of the denoising trajectory.

Existing approaches differ primarily in how they optimize towards the terminal reward. *Reward backpropagation* (Lee et al., 2023; Wu et al., 2024; Clark et al., 2024; Prabhudesai et al., 2023) differentiates the reward through the denoising chain. This provides direct, low-variance gradients, but requires storing the full trajectory in memory and is vulnerable to reward over-optimization, in which the generator exploits the reward model at the expense of perceptual quality and diversity (Clark et al., 2024; Prabhudesai et al., 2023). *Online reinforcement learning (RL)* (Black et al., 2023; Liu et al., 2025; Xue et al., 2025; Zheng et al., 2026) instead formulates denoising as a sequential policy and optimizes it with rollout-level policy gradients. By optimizing against sampled outcomes rather than differentiated rewards, this approach preserves diversity and mitigates over-optimization. However, relying on a single terminal scalar introduces high-variance gradients and poor sample efficiency.

Both paradigms inherently suffer from a *temporal credit-assignment* problem. Because the delayed terminal reward cannot be attributed to individual denoising steps, the training objective fails to identify which trajectory decisions drive the final improvement. Furthermore, lacking a step-wise signal, these methods cannot directly steer the sampling process at test time.

We argue that preference reward should instead be extracted where it is generated, along the denoising trajectory rather than at its endpoint. Our key insight is that the internal representations of a Diffusion Transformer (DiT) naturally encode early evidence of the trajectory quality. Even under high

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.

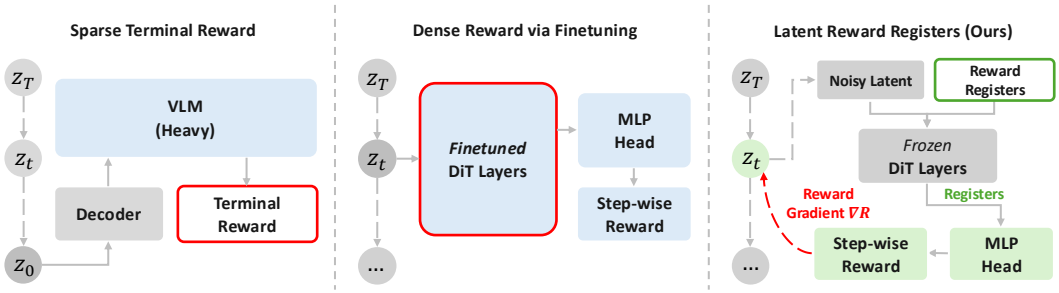

Fig. 1: **Latent Reward Registers vs. existing paradigms.** Unlike conventional alignment methods that rely on delayed, sparse feedback from a clean image (Left) or require intrusive fine-tuning of the generator (Middle), our approach (Right) prepends learnable registers to a frozen DiT. This enables the direct estimation of step-wise rewards from intermediate noisy latents (z_t). Consequently, we derive dense, differentiable reward gradients (∇R) for generation while keeping the pretrained generative dynamics fully intact.

noise levels, intermediate hidden states reflect the emerging global layout, prompt alignment, and visual structure. If the expected terminal reward can be read directly from any intermediate noisy latent, sparse endpoint feedback can be converted into a dense, step-wise reward signal, thereby bypassing the credit-assignment bottleneck.

Building on this insight, we introduce *Latent Reward Registers*, a lightweight mechanism that converts a frozen DiT into a step-wise reward model (Fig. 1). We prepend a small set of learnable, position-free register tokens to the noisy input sequence. These registers are trained to aggregate reward-relevant evidence from intermediate representations. A readout head then maps the aggregated evidence to the expected terminal preference at any given denoising step. Crucially, because this readout is differentiable with respect to the latent state, it provides a local reward gradient at negligible computational cost, explicitly indicating how intermediate latents should be perturbed to maximize the expected terminal preference.

This dense, differentiable reward signal supports alignment at both training and inference time. For training, *Reward-Gradient On-Policy Distillation* (RG-OPD) converts the register gradients into per-step optimization targets along the generator's own rollouts, replacing rollout-level policy gradients with local, low-variance supervision. For inference, *Reward-Guided Sampling* (RGS) applies magnitude-matched reward-gradient corrections during sampling, requiring neither parameter updates nor image decoding, and allowing multiple reward objectives to be composed at test time.

Experiments on SD3-Medium and FLUX.1-dev show that the registers predict terminal preference accurately even at high noise levels, attaining the highest pairwise accuracy among the evaluated latent reward models (Section 4.1). Building on this signal, RG-OPD surpasses online RL baselines in alignment quality while reducing GPU hours by up to 33× (Fig. 6). Furthermore, RGS significantly outperforms existing training-free sampling methods, achieving a superior reward–quality trade-off (Section 4.3). Our contributions are summarized as follows:

- **Latent reward readout from frozen generators.** We propose Latent Reward Registers, which estimate the expected terminal preference directly from intermediate noisy latents of a frozen DiT. At high noise levels, the registers attain the highest pairwise accuracy among latent reward models.
- **Training-time reward gradient on-policy distillation.** We show that register-derived reward gradients can replace rollout-level policy gradients: RG-OPD surpasses online RL baselines in alignment quality on SD3-Medium and FLUX.1-dev while reducing GPU hours by up to 33×.
- **Training-free reward gradient guidance.** We demonstrate that the register gradients also steer sampling directly: RGS improves target rewards without parameter updates and achieves a better reward–quality trade-off than existing training-free sampling methods.

2 Related Work

Diffusion Preference Alignment. Preference alignment for text-to-image generators can be applied at training time or inference time. Training-time methods can be categorized into three fami-

lies: reward backpropagation differentiates through the denoising chain to optimize a reward model (Lee et al., 2023; Clark et al., 2024; Prabhudesai et al., 2023), online reinforcement learning treats the denoising process as a policy and optimizes it with policy gradients (Liu et al., 2025; Zheng et al., 2026), and direct preference optimization (DPO) trains the generator on pairwise preferences without explicit reward modeling (Wallace et al., 2023). All three methods derive reward only from the final clean sample, and each pays a different cost. Reward backpropagation provides direct optimization signal, but its gradients must pass through the full denoising chain and fade before reaching the early steps that decide global structure and semantics. Online RL mitigates reward hacking by exploring the model's own distribution, but involves high variance and poor sample efficiency. The offline objective of DPO becomes biased once the generator drifts from the original policy, and its terminal reward is broadcast uniformly across denoising steps. Inference-time alignment avoids retraining, but existing schemes are computationally expensive, relying either on iterative search and optimization (Yeh et al., 2025; Tang et al., 2025) or on guidance gradients that become unstable at high noise levels (Song et al., 2023). Our method instead extracts a dense, differentiable, step-wise reward in the latent space, making alignment efficient at both training and inference time.

Latent Reward Models. Most reward models operate on clean, decoded images (Xu et al., 2023; Kirstain et al., 2023; Ma et al., 2025). Applying them to intermediate states would require repeated expensive decoding, and the resulting signals are unreliable on heavily corrupted inputs. Recent latent reward models instead evaluate noisy states directly, drawing evidence from selected internal features (Zhang et al., 2025), early cross-attention maps (Cui et al., 2026), separate process reward models (Mi et al., 2025; Go et al., 2026), or auxiliary heads that predict step-wise rewards (Yang et al., 2024). Diffusion-native variants go further, training noise-aware scorers inside the generator itself (Liu et al., 2026). Existing methods tend to rely on hand-picked features or additional value networks with substantial compute. Latent Reward Registers instead prepend lightweight, position-free register tokens to a frozen DiT, building a differentiable reward interface without hand-crafted feature heuristics and without altering the base generator, which directly supports alignment in both training and inference.

3 METHOD

Latent Reward Registers read frozen DiT features to score preferences at intermediate noisy states (Fig. 2). Their gradients define a reward-guided transition that RG-OPD distills into a student during training and RGS applies directly during sampling.

3.1 PRELIMINARIES AND NOTATION

Let $z_0 = \mathrm{Enc}(x)$ be the latent of image x under a frozen VAE with decoder Dec (Rombach et al., 2022), and let c be a text prompt. The flow ODE $\mathrm{d}z_t/\mathrm{d}t = v(z_t, t, c)$ runs from noise ($t = 1$) to data ($t = 0$) (Lipman et al., 2023), with $z_t \in \mathbb{R}^D$. For a decreasing schedule $\tau_0 = 1 > \cdots > \tau_N = 0$ and $\Delta\tau_i = \tau_{i+1} - \tau_i < 0$, the frozen reference and trainable student have Euler transitions

$$\mu_{s,i}(z, c) = z + \Delta\tau_i \, v_s(z, \tau_i, c), \qquad s \in \{\mathrm{ref}, \theta\}, \tag{1}$$

including native text guidance in v_s. Student rollouts start at $z_{\tau_0} \sim \mathcal{N}(0, I_D)$ and apply $z_{\tau_{i+1}} = \mu_{\theta,i}(z_{\tau_i}, c)$. Reward-register training instead uses forward-corrupted data:

$$z_t^{(a)} = (1 - t) z_0^{(a)} + t\epsilon, \qquad \epsilon \sim \mathcal{N}(0, I_D). \tag{2}$$

Both images in a preference pair (a, b) share t and ϵ.

3.2 REWARD REGISTER MECHANISM

For each latent state, we initialize K learnable, position-free registers $Q_t^{(0)} = Q^{(0)} \in \mathbb{R}^{K \times d_h}$, where d_h is the backbone hidden width. They accumulate information across the first L frozen DiT blocks through

$$Q_t^{(\ell+1)} = Q_t^{(\ell)} + \gamma_t^{(\ell)} \odot \mathrm{Attn}\Big(\bar{Q}_t^{(\ell)} W_Q^{\phi}, \mathbf{K}_t^{(\ell)}, \mathbf{V}_t^{(\ell)}\Big) \, W_O^{(\ell)}. \tag{3}$$

Here $\bar{Q}$ is the time-conditioned normalized register state; keys $\mathbf{K}$ and values $\mathbf{V}$ combine register, image, and text features. The query projection W_Q^{ϕ} is shared and trainable, while key, value, output,

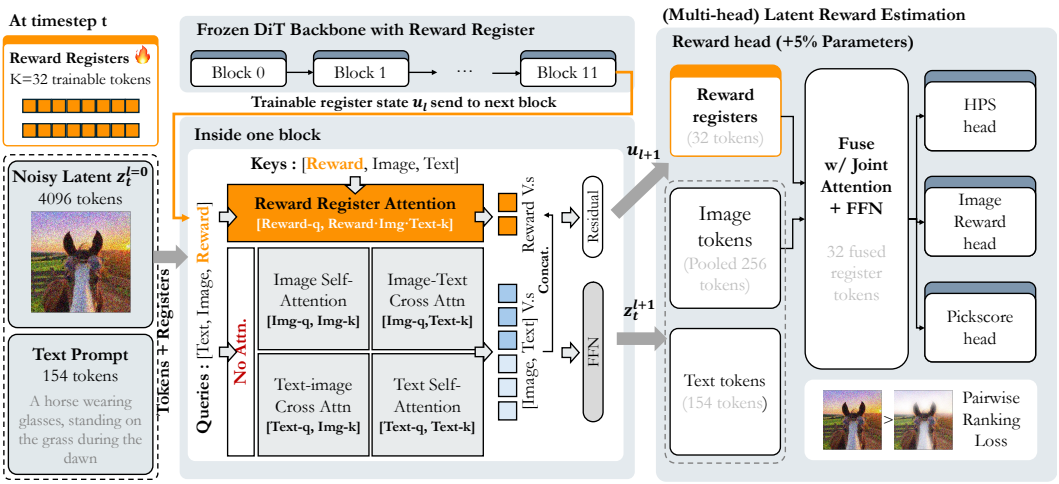

Fig. 2: **Overview of the Latent Reward Registers mechanism.** Learnable registers form a non-intrusive side stream through a frozen DiT. The registers read frozen image and text features through an auxiliary read path, thereby preserving the original velocity field. A lightweight fusion module combines the final register states with intermediate backbone features for multi-head latent reward estimation under pairwise ranking supervision.

normalization, and time-conditioning modules are reused from the frozen backbone. The gate $\gamma_t^{(\ell)}$ modulates the update element-wise. Registers are the only queries in this added attention operation; native image/text attention retains its original mask and never consumes register keys or values, and the register states are not concatenated back into the native residual stream. Native tokens therefore never read the registers, which also bypass the block FFNs; the backbone hidden states and velocity field remain unchanged. A lightweight attention readout fuses the registers with pooled features from selected blocks $\mathcal{B}$, producing $U_t \in \mathbb{R}^{K \times d_h}$. Full attention definitions and backbone-specific details appear in Appendix F.1 (Eq. 14).

Reward prediction and learning. Each reward-specific two-layer MLP $h_\phi^{(m)}$ predicts

$$r_\phi^{(m)}(z_t, c, t) = \frac{1}{K} \sum_{k=1}^{K} h_\phi^{(m)}(U_{t,k}), \qquad m = 1, \ldots, M. \tag{4}$$

We train each head to reproduce the pairwise rankings induced by its corresponding endpoint reward model. We use a noise-aware Thurstone objective (Thurstone, 1927), with the comparison variance increasing with t according to the adjustment used in DiNa-LRM (Liu et al., 2026). Only the registers, query projection, fusion readout, and reward heads are optimized; their exponential moving averages are used for evaluation and guidance. The resulting score is a differentiable proxy for preference ranking, rather than a calibrated reward estimate. Appendix B discusses its relation to diffused rewards, and Appendix F.5 (Table 9) gives the training settings.

Shared reward direction. At a trajectory state, each head provides

$$g_i^{(m)} = \nabla_{z_{\tau_i}} r_\phi^{(m)}(z_{\tau_i}, c, \tau_i). \tag{5}$$

Define $\mathrm{RMS}(u) = \sqrt{\|u\|_2^2 / D}$ and $\mathrm{unitRMS}(u) = u / \max\{\mathrm{RMS}(u), \delta\}$ for a numerical floor $\delta > 0$. For selected heads $\mathcal{M} \subseteq \{1, \ldots, M\}$, we use

$$g_i^{\mathcal{M}} = \sum_{m \in \mathcal{M}} \mathrm{unitRMS}(g_i^{(m)}), \qquad \widehat{g}_i^{\mathcal{M}} = \mathrm{unitRMS}(g_i^{\mathcal{M}}). \tag{6}$$

Each head thus contributes equally before aggregate normalization; a singleton $\mathcal{M}$ gives single-objective guidance.

3.3 Reward-Gradient On-Policy Distillation

RG-OPD aims to combine the optimization strength and efficiency of direct reward gradients with the resistance to reward over-optimization sought by online RL. Rather than differentiating a ter-

minal reward through a complete denoising chain, it converts latent reward gradients into local transition targets. These targets are rebuilt at states sampled from the current student, following on-policy distillation (Li et al., 2026; Yuan et al., 2026; Jiang et al., 2026), so supervision tracks the distribution being optimized.

At each student-visited state z_{τ_i}, we evaluate the frozen reference transition in Eq. 1 and its displacement

$$\Delta_{\mathrm{ref},i} = \mu_{\mathrm{ref},i}(z_{\tau_i}, c) - z_{\tau_i}. \tag{7}$$

For strength $\alpha_i \geq 0$, the reward-guided target is

$$\mu_{\mathrm{tar},i}(z_{\tau_i}, c) = \mu_{\mathrm{ref},i}(z_{\tau_i}, c) + \alpha_i \, \mathrm{RMS}(\Delta_{\mathrm{ref},i}) \, \widehat{g}_i^{\mathcal{M}}. \tag{8}$$

The reference transition anchors each target to the pretrained dynamics, while RMS normalization controls the reward correction relative to the solver displacement. Setting $\alpha_i = 0$ recovers the reference transition as a retention target; no separate teacher network is learned. Together, on-policy target refresh and reference anchoring are intended to limit reward-driven drift while retaining a direct optimization signal. This is a design for mitigating reward hacking, not a guarantee against errors in the learned preference proxy. Our experiments assess the resulting reward–quality trade-off against reward-backpropagation and online RL baselines (Table 2 and Fig. 4).

Let sg denote stop-gradient, p_θ the current student's trajectory distribution, $\mathcal{C}$ the prompt distribution, and $\mathcal{J} \subseteq \{0, \ldots, N-1\}$ the optimized steps. We minimize

$$\mathcal{L}_{\text{RG-OPD}}(\theta) = \mathbb{E}_{c\sim\mathcal{C}} \mathbb{E}_{(z_{\tau_0},\ldots,z_{\tau_N})\sim p_\theta(\cdot|c)} \left[\frac{1}{|\mathcal{J}|D} \sum_{i\in\mathcal{J}} \left\| \mu_{\theta,i}(\mathrm{sg}(z_{\tau_i}), c) - \mathrm{sg}(\mu_{\mathrm{tar},i}(z_{\tau_i}, c)) \right\|_2^2 \right]. \tag{9}$$

States and targets, including the reward direction, are detached, so the update is a local semi-gradient step with no differentiation through p_θ or the denoising rollout. Only θ is updated; the reference model and reward-register parameters remain frozen. This avoids backpropagation through full trajectories and replaces terminal-score policy-gradient estimation with per-step regression targets. Appendix F.3 gives the complete algorithm (Algorithm 1).

3.4 Reward-Guided Sampling

RGS uses the latent reward gradient to steer sampling without updating the generator or reward register. At each state, the frozen generator first proposes $\mu_{\mathrm{ref},i}$; the selected reward heads then provide the normalized direction $\widehat{g}_i^{\mathcal{M}}$. We add a correction of magnitude $\alpha_i \, \mathrm{RMS}(\Delta_{\mathrm{ref},i})$ and use the resulting target from Eq. 8 directly as the next latent, $z_{\tau_{i+1}} = \mu_{\mathrm{tar},i}(z_{\tau_i}, c)$. Unlike RG-OPD, RGS requires no student or distillation: the reward correction is applied on the sampled trajectory itself. The displacement scaling makes α_i a relative guidance strength, and $\alpha_i = 0$ recovers the reference sampling step.

Guidance scheduler. Early corrections can influence global structure, whereas corrections near the data endpoint bring smaller gains and can distort fine details. We therefore use three flow-time bands: $\alpha_i = \alpha_{\mathrm{early}}$ for $\tau_i > \tau_{\mathrm{hi}}$, $\alpha_i = \alpha_{\mathrm{mid}}$ for $\tau_{\mathrm{lo}} < \tau_i \leq \tau_{\mathrm{hi}}$, and $\alpha_i = 0$ for $\tau_i \leq \tau_{\mathrm{lo}}$, with $\alpha_{\mathrm{early}} > \alpha_{\mathrm{mid}} \geq 0$. This concentrates guidance on early and intermediate states while leaving the low-noise tail to the reference generator. Table 10 lists the backbone-specific settings. Algorithm 2 gives the full procedure, and Appendix F.2 describes the extension beyond Euler integration.

4 Experiments

We evaluate latent reward prediction, RG-OPD training efficiency, and RGS inference effectiveness on SD3-Medium, with FLUX.1-dev providing cross-architecture evidence. Appendix A indexes supplementary content; Appendix F collects experimental settings.

4.1 Latent Reward Prediction

Setup. We evaluate pairwise preference accuracy on noisy latents with the frozen SD3-Medium backbone; the complete datasets, corruption protocol, and baseline list are given in Appendix F.4.

Table 1: **High-noise pairwise preference accuracy (%) on four benchmarks.** Latent reward models score noisy states at t=0.8, whereas endpoint reward models score clean images. **Red bold** and blue underline mark the best and second-best results within each model category.

Model	Backbone	Pairwise Preference Accuracy (%)				
		ImageReward	HPDv2	HPDv3	GenAI-Bench	Avg
Endpoint Reward Models						
ImageReward	CLIP	65.15	73.95	58.74	63.41	65.31
PickScore	CLIP	62.73	79.44	65.67	69.98	69.45
HPSv2	CLIP	65.62	82.58	64.69	67.62	70.13
HPSv3	Qwen2VL-7B	**67.03**	**85.36**	**76.03**	**70.95**	**74.84**
Latent Reward Models						
LRM-SD1.5	SD-1.5	55.88	64.80	52.37	52.46	56.38
LRM-SDXL	SDXL	58.13	66.49	52.87	54.67	58.04
PAVRM	SD3-Medium	56.76	73.52	68.18	57.20	63.91
Diffusion Probe	SD3-Medium	58.67	75.29	68.68	57.14	64.95
DiNa-LRM	SD3-Medium	60.10	**78.76**	68.08	58.94	66.47
Reward Register	SD3-Medium					
+HPS		61.84	78.46	**69.96**	61.41	67.92
+ImageReward		**63.32**	75.73	63.20	62.61	66.22
+Raw score-sum		63.12	78.10	69.10	**62.95**	**68.32**

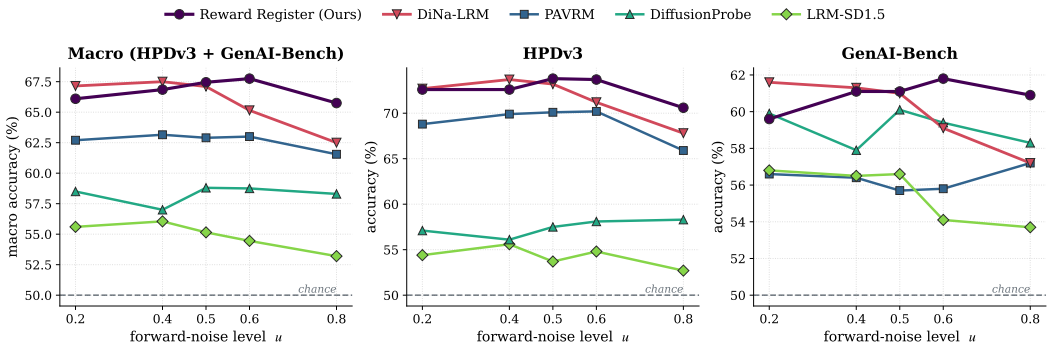

Fig. 3: **Preference accuracy across noise levels.** Pairwise preference accuracy of latent reward models evaluated at noise level t. The reward register retains accuracy into the high-noise regime, whereas DiNa-LRM degrades sharply as the latents become less informative.

Results. The register outperforms all latent baselines on three of four benchmarks despite heavily corrupted inputs (Table 1). Clean-image models provide a reference under their native input condition. Appendix C reports FLUX.1-dev transfer (Appendix C.1, Table 5) and endpoint rank agreement (Appendix C.2, Fig. 7).

4.2 Reward-Gradient On-Policy Distillation

Setup. RG-OPD, reward-backpropagation, and online RL use matched evaluations, targeting HPSv3 or ImageReward and assessing perceptual quality with MUSIQ and CLIP-IQA. Appendix F.7 gives training details (Table 11).

Results. RG-OPD attains the highest HPSv3 scores on both architectures with a balanced multi-reward profile (Table 2). Reward-backpropagation baselines often maximize target metrics at the expense of perceptual quality. Under matched prompts and seeds, RG-OPD preserves prompt-relevant content without the artifacts exhibited by several baselines (Fig. 4); Appendix D.1 provides further examples (Fig. 8).

Table 2: **Training-time preference alignment on SD3-Medium and FLUX.1-dev.** HPSv3 and ImageReward are optimization targets, MUSIQ and CLIP-IQA measure no-reference perceptual quality. **Red bold** and blue underline mark the best and second-best results within each category.

Method	SD3-Medium				FLUX.1-dev			
	HPSv3↑	IR↑	MUSIQ↑	CIQA↑	HPSv3↑	IR↑	MUSIQ↑	CIQA↑
Baselines								
Base + CFG	8.55	1.183	73.2	0.694	9.89	1.142	72.9	0.669
ReFL +HPS	10.84	1.341	67.0	0.512	12.00	1.337	56.8	0.428
ReFL +IR	9.69	**1.678**	**75.0**	0.716	9.31	1.293	62.7	0.425
AlignProp +HPS	11.20	1.472	68.1	0.580	11.83	1.241	59.4	0.481
AlignProp +IR	9.51	1.611	71.2	0.491	9.47	1.213	68.2	0.577
Flow-GRPO +HPS	10.26	1.349	74.3	0.692	12.06	1.212	**75.9**	0.697
Flow-GRPO +IR	9.45	1.297	74.1	0.705	11.13	**1.377**	75.7	0.702
Diffusion-NFT +HPS	9.51	0.995	72.4	0.551	11.54	1.056	75.1	0.695
Diffusion-NFT +IR	9.68	1.221	74.4	0.686	11.50	1.241	74.1	0.699
Latent Reward Registers with on-policy distillation (Ours)								
Latent Reward Registers (OPD) +HPS	11.57	1.336	74.9	**0.726**	**12.38**	1.229	74.4	0.648
Latent Reward Registers (OPD) +IR	10.98	1.444	72.1	0.674	10.82	1.367	75.6	**0.704**
Latent Reward Registers (OPD) +MH	**11.62**	1.417	74.0	0.704	11.91	1.346	75.0	0.681

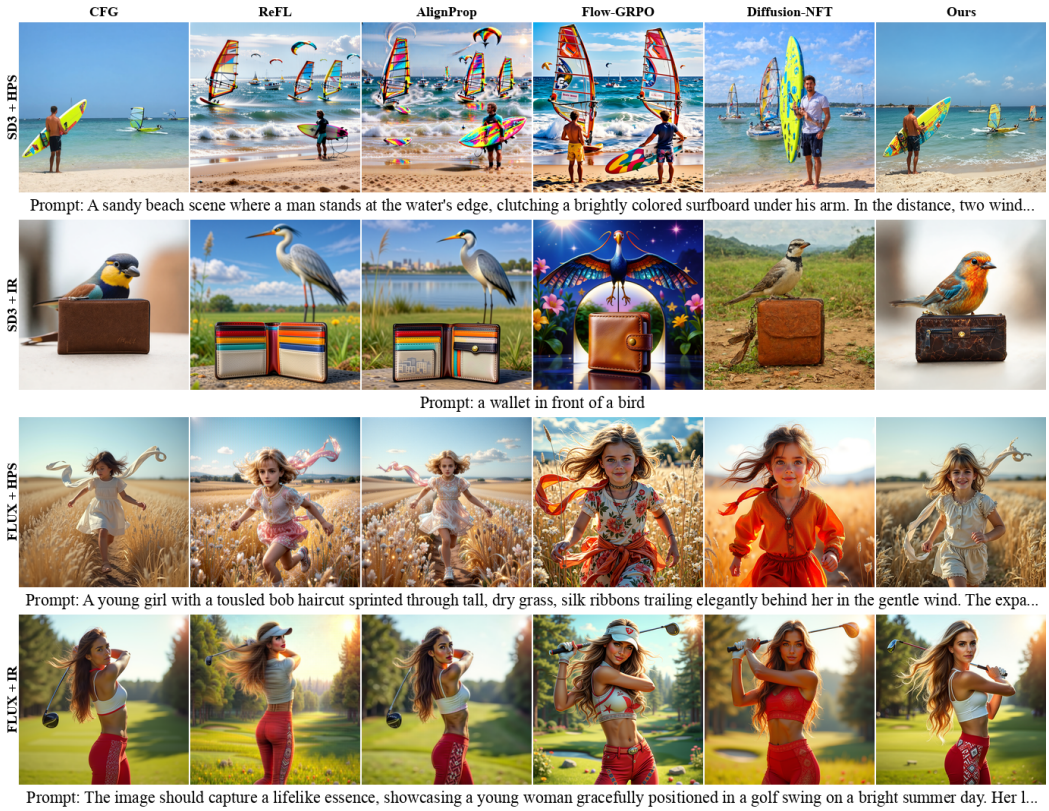

Fig. 4: **Qualitative comparison of training-time alignment.** RG-OPD is compared against CFG, ReFL, AlignProp, Flow-GRPO, and Diffusion-NFT under matched prompts and seeds. RG-OPD achieves better prompt alignment and higher perceptual quality without introducing visible artifacts.

4.3 Reward-Guided Sampling

Setup. RGS, DNO, and Demon keep the generator frozen and use matched evaluation conditions (Table 3); Appendix F.6 gives sampling settings.

Table 3: **Reward-Guided Sampling with frozen SD3-Medium and FLUX.1-dev generators.** RGS provides the strongest HPS alignment and a favorable reward–quality trade-off among the evaluated training-free methods. HPSv3 and ImageReward (IR) are optimization targets. MUSIQ and CLIP-IQA (CIQA) measure perceptual quality. DNO and Demon use HPSv3 for their `+HPS` variants. **Red bold** and blue underline mark the best and second-best results within each backbone.

	SD3-Medium				FLUX.1-dev			
Method	HPSv3↑	IR↑	MUSIQ↑	CIQA↑	HPSv3↑	IR↑	MUSIQ↑	CIQA↑
Baselines								
Base + CFG	8.55	1.183	73.2	0.694	9.89	1.142	72.9	0.669
DNO `+HPS`	8.85	1.272	73.1	0.698	9.64	1.191	71.9	0.655
Demon `+HPS`	9.09	1.311	73.0	0.685	9.56	1.209	68.2	0.605
DNO `+IR`	8.19	1.454	72.3	0.693	9.25	1.350	69.5	0.632
Demon `+IR`	8.90	**1.596**	72.5	0.674	9.27	**1.501**	68.2	0.603
Reward-Guided Sampling (Ours)								
Latent Reward Registers `+HPS`	**10.70**	1.347	**75.0**	**0.728**	**11.03**	1.212	73.3	0.715
Latent Reward Registers `+IR`	10.12	1.465	73.0	0.694	10.38	1.297	73.2	0.710
Latent Reward Registers `+MH`	10.48	1.413	74.1	0.723	10.80	1.276	**73.5**	**0.717**



Fig. 5: **Comparison of training-free sampling methods.** CFG, DNO, Demon, and our single- and multi-head RGS variants use matched prompts and seeds within each backbone.

Results. Each single-head RGS variant improves its target metric over CFG while preserving the evaluated perceptual metrics. RGS achieves the highest HPSv3 among the evaluated training-free methods and a favorable reward–quality trade-off under ImageReward optimization. Fig. 5 compares matched prompts and seeds. Appendix E.1 compares single- and multi-head guidance (Fig. 10); Appendix E.7 illustrates guidance-strength effects (Fig. 13).

Appendix E examines the magnitude-matched CFG-direction control (Appendix E.2, Table 6), inference cost (Appendix E.3, Table 7), sampling budgets and strengths (Appendix E.4, Fig. 12), reward trajectories (Appendix E.5, Fig. 11), and ODE solver robustness (Appendix E.6, Table 8).

4.4 Analysis

Register design. The persistent register state is the main driver of accuracy. Removing the register causes the largest performance drop, while the skip-FFN design and the trainable query projection each add complementary gains (Table 4). Reducing the number of readout layers, by contrast, has little effect, indicating that persistent side-stream aggregation matters more than deep multi-scale feature extraction.

Table 4: **Reward Register Design Ablation.** Pairwise preference accuracy evaluated across four benchmarks. Δ quantifies the deviation in average accuracy relative to the full deployed method. **Bold** indicates the optimal performance within each respective column.

Variant	Pairwise Preference Accuracy (%)					Δ
	ImageReward	HPDv2	HPDv3	GenAI-Bench	Avg	
Full method	**63.12**	78.10	69.10	**62.95**	**68.32**	—
w/o reward register	56.86	74.12	65.42	60.76	64.29	−4.03
w/o skip-FFN	58.15	77.82	68.54	60.38	66.23	−2.09
w/o trainable q-proj	58.20	77.62	68.62	60.42	66.21	−2.11
w/ layers $[4]$	59.39	79.32	69.57	61.69	67.49	−0.83
w/ layers $[4, 8]$	60.01	**79.90**	**70.35**	61.30	67.89	−0.43

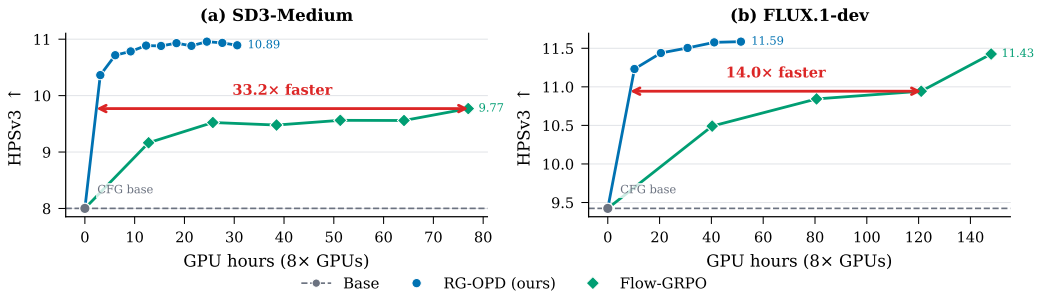

Fig. 6: **Training efficiency of RG-OPD.** HPSv3 optimization trajectories of RG-OPD and Flow-GRPO against aggregate GPU hours on SD3-Medium and FLUX.1-dev. RG-OPD reaches equivalent HPSv3 score while $14.0\times$–$33.2\times$ faster than Flow-GRPO in GPU hours, reflecting the dense, step-wise supervision provided by the latent reward registers.

Training efficiency. RG-OPD combines dense step-wise supervision with on-policy trajectories that track the evolving student. It reaches Flow-GRPO-equivalent HPSv3 levels $14.0\times$–$33.2\times$ faster in GPU hours on both backbones (Fig. 6). Appendix D includes reward-tilt analysis (Appendix D.2, Fig. 9).

Accuracy across noise levels. **The register stays reliable at high noise levels.** DiNa-LRM performs competitively at low noise but degrades sharply as inputs become uninformative, whereas the register retains accuracy throughout (Fig. 3), supporting its use as an early, step-wise guidance signal.

5 Conclusion

In this work, we show that intermediate DiT representations contain information about terminal preference early in the denoising process. Building on this observation, we propose Latent Reward Registers, a non-intrusive auxiliary stream that predicts endpoint preference from intermediate noisy latents. The resulting dense reward signal supports two alignment strategies. Reward-Gradient On-Policy Distillation (RG-OPD) uses step-wise gradient supervision to outperform online RL baselines in efficiency and alignment, while avoiding the over-optimization observed with direct reward backpropagation. Reward-Guided Sampling (RGS) applies magnitude-matched register gradients to steer generation without parameter updates, improving multiple target rewards while preserving the evaluated perceptual metrics. Overall, the framework addresses the temporal credit-assignment challenge with a practical latent preference model for diffusion alignment. Appendix G discusses future directions.

AI USE STATEMENT

In preparing this submission, we used generative AI tools to assist with manuscript editing, LaTeX cross-reference auditing, and formatting checks. The authors reviewed all AI-assisted work and take responsibility for the final content, claims, experiments, and conclusions of this paper.

REFERENCES

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*, 2023.

Kevin Clark, Paul Vicol, Kevin Swersky, and David Fleet. Directly fine-tuning diffusion models on differentiable rewards. In *International Conference on Learning Representations*, volume 2024, pp. 4793–4822, 2024.

Benlei Cui, Bukun Huang, Zhizeng Ye, Xuemei Dong, Tuo Chen, Hui Xue, Dingkang Yang, Longtao Huang, Jingqun Tang, and Haiwen Hong. Diffusion probe: Generated image result prediction using cnn probes, 2026. URL https://arxiv.org/abs/2602.23783.

Hyojun Go, Hyungjin Chung, Prune Truong, Goutam Bhat, Li Mi, Zhaochong An, Zixiang Zhao, Dominik Narnhofer, Serge Belongie, Federico Tombari, and Konrad Schindler. Stitched value model for diffusion alignment. *arXiv preprint arXiv:2605.19804*, 2026.

Ernst Hairer, Syvert P. Nørsett, and Gerhard Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems*, volume 8 of *Springer Series in Computational Mathematics*. Springer, 2nd revised edition, 1993. doi: 10.1007/978-3-540-78862-1.

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.

Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. MUSIQ: Multi-scale image quality transformer. In *IEEE/CVF International Conference on Computer Vision*, 2021.

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. *arXiv preprint arXiv:2305.01569*, 2023.

Kimin Lee, Hao Liu, Moonkyung Ryu, Olivia Watkins, Yuqing Du, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, and Shixiang Shane Gu. Aligning text-to-image models using human feedback. *arXiv preprint arXiv:2302.12192*, 2023.

Baiqi Li, Zhiqiu Lin, Deepak Pathak, Jiayao Li, Yixin Fei, Kewen Wu, Tiffany Ling, Xide Xia, Pengchuan Zhang, Graham Neubig, and Deva Ramanan. GenAI-Bench: Evaluating and improving compositional text-to-visual generation. *arXiv preprint arXiv:2406.13743*, 2024.

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. *arXiv preprint arXiv:2605.15055*, 2026.

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

Gongye Liu, Bo Yang, Yida Zhi, Zhizhou Zhong, Lei Ke, Didan Deng, Han Gao, Yongxiang Huang, Kaihao Zhang, Hongbo Fu, and Wenhan Luo. Beyond VLM-based rewards: Diffusion-native latent reward modeling. *arXiv preprint arXiv:2602.11146*, 2026.

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 (NeurIPS)*, 2025.

Yuhang Ma, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. HPSv3: Towards wide-spectrum human preference score. *arXiv preprint arXiv:2508.03789*, 2025.

Xiaoyue Mi, Wenqing Yu, Jiesong Lian, Shibo Jie, Ruizhe Zhong, Zijun Liu, Guozhen Zhang, Zixiang Zhou, Zhiyong Xu, Yuan Zhou, Qinglin Lu, and Fan Tang. Video generation models are good latent reward models. *arXiv preprint arXiv:2511.21541*, 2025.

Mihir Prabhudesai, Anirudh Goyal, Deepak Pathak, and Katerina Fragkiadaki. Aligning text-to-image diffusion models with reward backpropagation, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Jiaming Song, Qinsheng Zhang, Hongxu Yin, Morteza Mardani, Ming-Yu Liu, Jan Kautz, Yongxin Chen, and Arash Vahdat. Loss-guided diffusion models for plug-and-play controllable generation. In *International Conference on Machine Learning*, 2023.

Zhiwei Tang, Jiangweizhi Peng, Jiasheng Tang, Mingyi Hong, Fan Wang, and Tsung-Hui Chang. Tuning-free alignment of diffusion models with direct noise optimization. In *International Conference on Machine Learning*, 2025.

Louis L. Thurstone. A law of comparative judgment. *Psychological Review*, 34(4):273–286, 1927.

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. *arXiv preprint arXiv:2311.12908*, 2023.

Jianyi Wang, Kelvin C. K. Chan, and Chen Change Loy. Exploring CLIP for assessing the look and feel of images. In *AAAI Conference on Artificial Intelligence*, 2023.

Junyi Wu, Weijian Luo, Haoyang Zheng, Ruizhe Zhang, and Guang Lin. Diff-instruct with diffused reward: Towards principled one-step generator rl. *arXiv preprint arXiv:2605.24001*, 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.

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 (ECCV)*, 2024.

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. *arXiv preprint arXiv:2304.05977*, 2023.

Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, et al. DanceGRPO: Unleashing GRPO on visual generation. *arXiv preprint arXiv:2505.07818*, 2025.

Shentao Yang, Tianqi Chen, and Mingyuan Zhou. A dense reward view on aligning text-to-image diffusion with preference. *arXiv preprint arXiv:2402.08265*, 2024.

Po-Hung Yeh, Kuang-Huei Lee, and Jun-Cheng Chen. Training-free diffusion model alignment with sampling demons. In *International Conference on Learning Representations*, 2025.

Qianhao Yuan, Jie Lou, Xing Yu, Hongyu Lin, Le Sun, Xianpei Han, and Yaojie Lu. Vision-OPD: Learning to see fine details for multimodal LLMs via on-policy self-distillation. *arXiv preprint arXiv:2605.18740*, 2026.

Tao Zhang, Cheng Da, Kun Ding, Huan Yang, Kun Jin, Yan Li, Tingting Gao, Di Zhang, Shiming Xiang, and Chunhong Pan. Diffusion model as a noise-aware latent reward model for step-level preference optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

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 (ICLR)*, 2026.

A Appendix Overview

This appendix contains supplementary material: the experimental protocols, extended results for Latent Reward Registers, training-time alignment, and Reward-Guided Sampling (RGS), and directions for future research.

1. **Latent reward potentials and diffused rewards** (Appendix B)
2. **Additional Reward-Register Results** (Appendix C)
   - Pairwise preference accuracy (Appendix C.1)
   - Rank agreement with endpoint reward models (Appendix C.2)
3. **Additional Training-Time Alignment Results** (Appendix D)
   - Qualitative results (Appendix D.1)
   - Effect of reward tilt on training dynamics (Appendix D.2)
4. **Additional Reward-Guided Sampling Results** (Appendix E)
   - Multi-head RGS (Appendix E.1)
   - Magnitude-matched CFG-direction control (Appendix E.2)
   - Inference cost (Appendix E.3)
   - Sampling budgets and guidance strengths (Appendix E.4)
   - Reward along the guided trajectory (Appendix E.5)
   - Robustness to the ODE solver (Appendix E.6)
   - Qualitative comparison across prompts (Appendix E.7)
5. **Experimental Settings** (Appendix F)
   - Reward register implementation (Appendix F.1)
   - Higher-order solver extension (Appendix F.2)
   - Complete algorithms (Appendix F.3)
   - Reward register training (Appendix F.5)
   - Reward-gradient on-policy distillation (Appendix F.7)
   - Reward-Guided Sampling (Appendix F.6)
6. **Future Work** (Appendix G)

B Latent Reward Potentials and Their Connection to Diffused Rewards

This section relates direct latent reward prediction to the diffused-reward construction of DIDR (Wu et al., 2026). Let $q_0(z_0 \mid c)$ denote the reference endpoint distribution and $R(z_0, c)$ a terminal reward. For temperature $\eta > 0$, define $q_0^\star(z_0 \mid c) = Z(c)^{-1} q_0(z_0 \mid c) \exp(R(z_0, c)/\eta)$. Propagating both endpoint distributions through the same forward corruption kernel gives

$$q_t^\star(z_t \mid c) = \frac{q_t(z_t \mid c)}{Z(c)} \mathbb{E}_q[\exp(R(z_0, c)/\eta) \mid z_t, c]. \tag{10}$$

Define $A_t(z_t, c) := \eta \log \mathbb{E}_q[\exp(R/\eta) \mid z_t, c]$. Then the reward-induced score correction is

$$s_R = \nabla_{z_t} \log q_t^\star - \nabla_{z_t} \log q_t = \eta^{-1} \nabla_{z_t} A_t. \tag{11}$$

If a latent reward model is trained by squared-error regression under the same joint distribution, its population optimum is $M_t(z_t, c) := \mathbb{E}_q[R \mid z_t, c]$. Writing $\kappa_{n,t}$ for conditional cumulants, $\kappa_{1,t} = M_t$ and $\kappa_{2,t} = \mathrm{Var}(R \mid z_t, c)$, the cumulant expansion gives

$$A_t = M_t + \frac{\kappa_{2,t}}{2\eta} + \frac{\kappa_{3,t}}{6\eta^2} + O(\eta^{-3}), \tag{12}$$

so, when differentiable,

$$s_R = \eta^{-1} \nabla M_t + (2\eta^2)^{-1} \nabla \kappa_{2,t} + O(\eta^{-3}). \tag{13}$$

Thus conditional-mean reward guidance is the leading-order diffused-reward correction; higher-order terms account for state-dependent reward uncertainty.

For a fixed deterministic sampler with continuation map $F_{t\to 0}$, the point-mass continuation distribution satisfies $\widetilde{A}_t = \widetilde{M}_t = R(F_{t\to 0}(z_t, c), c)$. This is the degenerate single-candidate ($K = 1$) case.

Table 5: **High-noise pairwise preference accuracy (%) on FLUX.1-dev.** Register variants on a frozen FLUX.1-dev backbone, evaluated at t=0.8 under the protocol of Table 1. **Bold** marks the best result in each column.

Variant	ImageReward	HPDv2	HPDv3	GenAI-Bench	Avg
+HPS	62.12	81.18	**73.30**	61.75	69.59
+ImageReward	63.13	78.84	65.86	**63.42**	67.81
+Raw score-sum	**63.46**	**81.20**	72.36	62.96	**70.00**

It is distinct from the forward-corruption posterior $q(z_0 \mid z_t, c)$, which is generally non-degenerate even when inference uses a deterministic ODE. DIDR's multi-candidate estimator further applies reward-softmax weights to differentiable denoising continuations.

Our registers use pairwise ranking rather than squared-error supervision. Their outputs are therefore differentiable preference proxies, not calibrated estimates of M_t, A_t, or an exact DIDR score. The derivation motivates intermediate-latent guidance, while the RMS normalization and displacement matching in RG-OPD and RGS define practical updates rather than exact reward-tilted marginal dynamics.

C Additional Reward-Register Results

C.1 Pairwise Preference Accuracy

Pairwise preference accuracy measures how often a model scores the human-preferred image higher within a held-out pair. The evaluation spans the ImageReward, HPDv2, HPDv3, and GenAI-Bench datasets, with 54,170 pairs in total. Endpoint reward models are evaluated on fully decoded, pristine images, while latent reward models score pairs that have been forward-noised to t=0.8. Across both backbones, +Raw score-sum denotes an unweighted ensemble of the constituent register heads.

On FLUX.1-dev the raw score-sum ensemble reaches an average accuracy of 70.00% (Table 5), confirming that the register mechanism transfers to an alternative frozen backbone.

C.2 Rank Agreement with Endpoint Reward Models

Each register head preserves the ranking behavior of its corresponding endpoint reward model at high noise. Fig. 7 compares the ranking percentiles the register assigns to noised latents with those of the endpoint models on fully denoised images, over a sample of 800 images. The Spearman correlation ρ ranges from 0.76 to 0.83 and stays stable across noise levels $t \in \{0.2, 0.5, 0.8\}$.

D Additional Training-Time Alignment Results

D.1 Qualitative Results

Fig. 8 provides additional qualitative examples of training-time alignment. Each row uses the same text prompt and initial latent seed, so the comparison reflects consistent gains in prompt adherence and visual quality across prompts.

D.2 Effect of Reward Tilt on Training Dynamics

Fig. 9 further compares training dynamics under two reward tilts (α=0.20 vs. 0.40): the larger tilt accelerates convergence and raises peak performance, and the internal register reward tracks the endpoint HPSv2 evaluation throughout training.

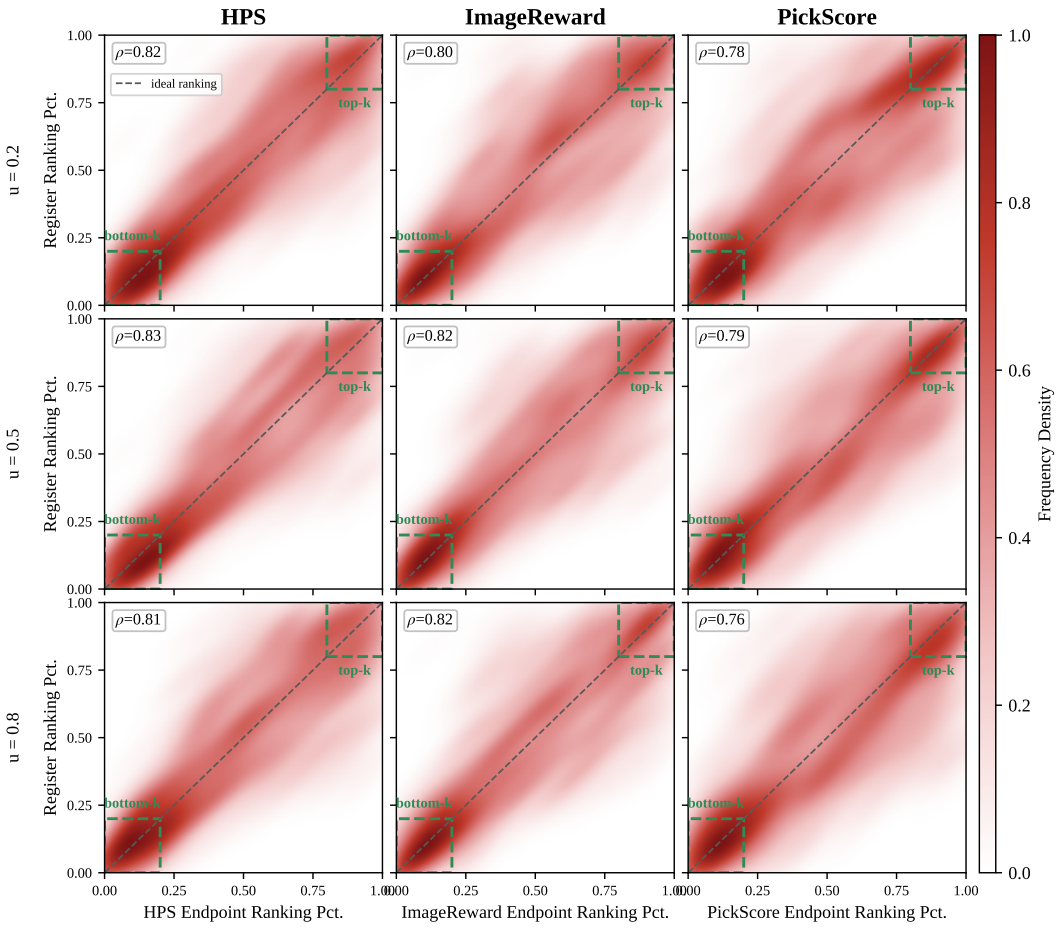

Fig. 7: **Rank agreement between register heads and endpoint reward models.** Joint density of ranking percentiles from register heads on noised latents versus endpoint models on fully decoded images, across noise levels $t \in \{0.2, 0.5, 0.8\}$. ρ denotes the Spearman rank correlation. The diagonal marks perfect rank preservation.

E Additional Reward-Guided Sampling Results

E.1 Multi-Head Reward-Guided Sampling

Multi-reward RGS balances competing objectives. Single-head guidance targets one reward, whereas the multi-head variant maintains a balanced profile across reward and perceptual metrics, consistent with the trends in Table 3 and Fig. 10. This supports the use of equal-RMS aggregation for multi-objective composition.

E.2 Magnitude-Matched CFG-Direction Control

To separate the effect of correction magnitude from correction direction, the CFG Control baseline replaces the RGS reward gradient with the SD3 conditional-unconditional residual. At each active step the applied correction is $\alpha_i \, \mathrm{RMS}(\Delta_{\mathrm{ref},i}) \, \mathrm{unitRMS}(\Delta\tau_i \, (v_{\mathrm{cond}} - v_{\mathrm{uncond}}))$, following the same α_i schedule as RGS. The two controls therefore share the per-step magnitude-matching rule and differ only in the direction applied.

CFG Control improves some perceptual and reward metrics relative to standard CFG but does not approach the alignment of RGS (Table 6). Since the correction magnitude is identical, this comparison isolates the learned reward direction as the driver of the improvements.

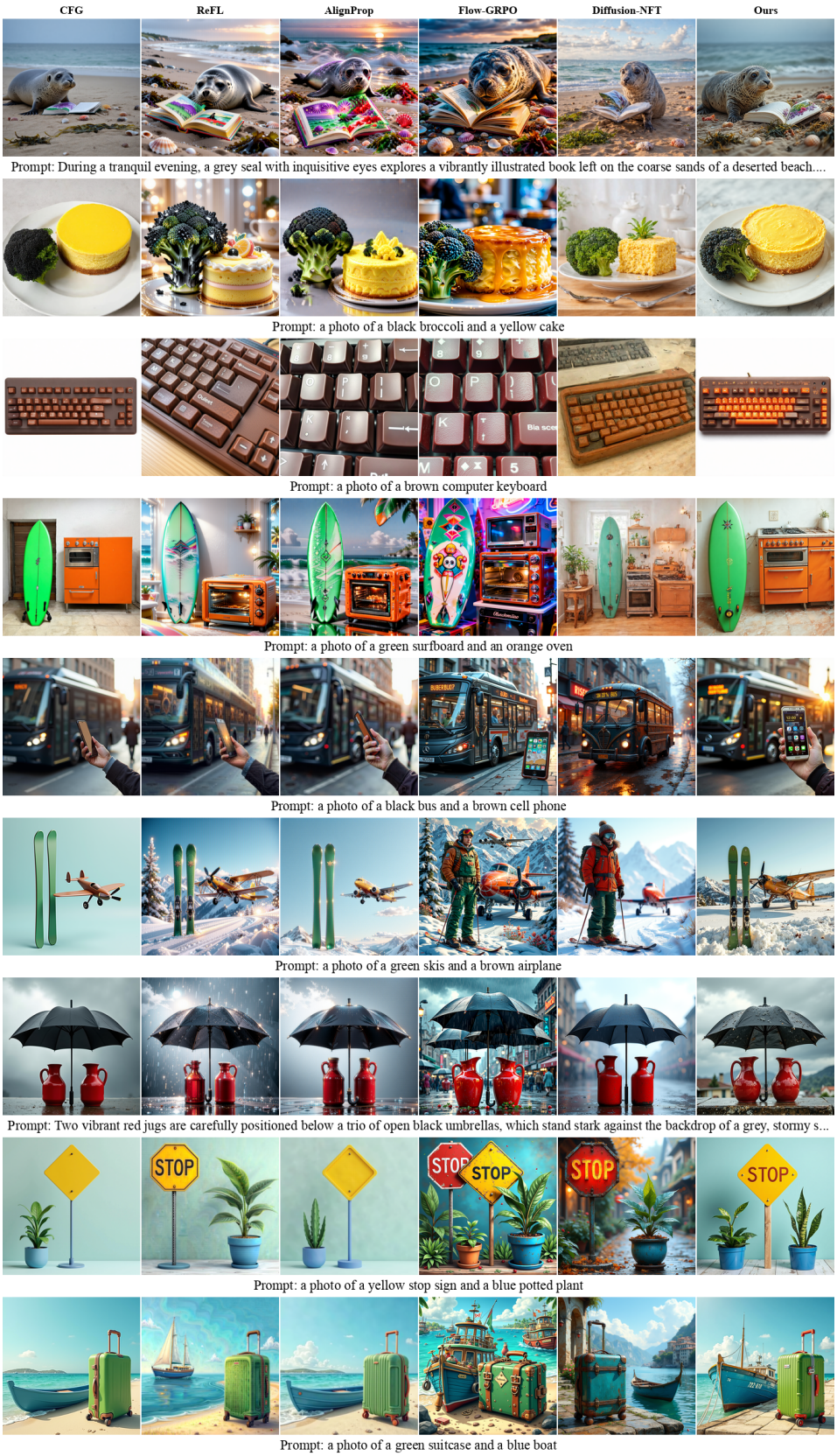

Fig. 8: **Extended qualitative comparison of training-time alignment.** We provide additional visual examples comparing our RG-OPD framework against baseline methods across a diverse set of text prompts.

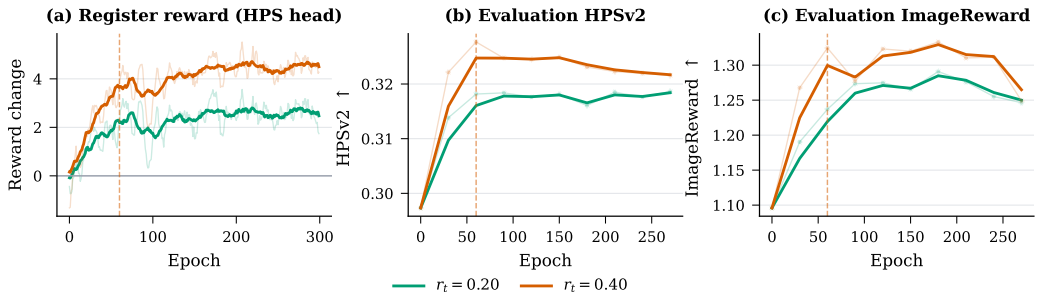

Fig. 9: **Effect of the reward tilt α during RG-OPD training.** Training dynamics are compared between α=0.20 and α=0.40 (denoted r_t in the panel labels). The results show that a larger reward scale accelerates convergence and yields a higher peak performance. Moreover, the upward trend of the internal latent reward (a) remains consistent with the endpoint HPSv2 evaluation (b), indicating alignment between the intermediate proxy and the terminal objective. ImageReward (c) is evaluated as an independent, unoptimized metric to monitor the overall generation quality during the process.

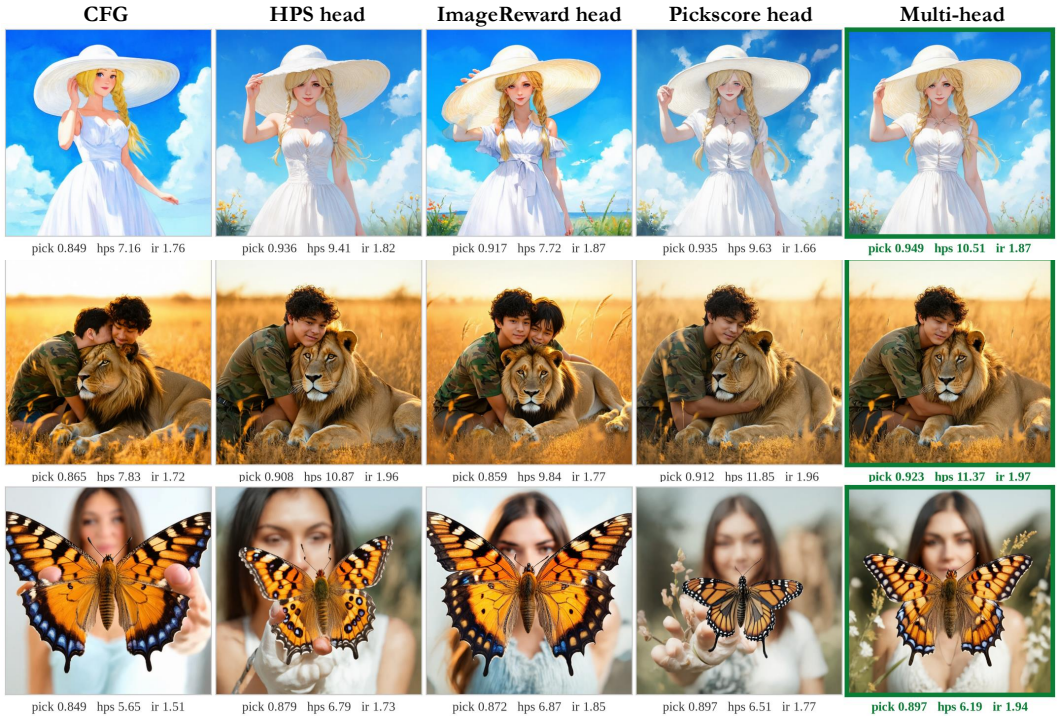

Fig. 10: **Multi-head RGS.** Multi-head reward gradients achieve a balanced performance across different reward metrics, compared to single-head RGS under identical settings.

E.3 Inference Cost of Reward-Guided Sampling

Latent Reward Registers has a memory footprint comparable to standard CFG and an inference latency competitive with the most efficient guided baselines (Table 7). All reported numbers are measured end-to-end on a single GPU in bf16, generating 1024^2 images over 42 steps with CFG scale 4.5 and batch size 1. Each method runs in its own process and is averaged over ten runs after ten warm-up runs. In the standard deployment the register shares the frozen generator backbone and text encoders of the sampling pipeline, so the memory overhead is limited to the 147.6 M trainable parameters of the read path and reward head: peak memory rises only from 16.9 GiB to 17.6 GiB versus unguided CFG (a $1.04\times$ factor). Baselines that require separate neural backbones instead consume 27–34 GiB. Computationally, each guided step adds roughly 50 ms and 5.9 TFLOPs on top of the 12.4 TFLOPs of a base generation step.

Table 6: **Magnitude-matched direction control on SD3-Medium.** CFG Control matches the per-step RMS magnitude of the RGS correction but orients the update along the standard CFG residual instead of the learned reward direction. Evaluation follows the matched prompt–seed protocol of Table 3. **Red bold** and blue underline mark the best and second-best results in each column.

Method	HPSv3↑	IR↑	MUSIQ↑	CIQA↑
Base + CFG	8.55	1.183	73.2	0.694
CFG Control	8.50	1.214	74.14	0.720
Latent Reward Registers +HPS	**10.70**	1.347	**75.0**	**0.728**
Latent Reward Registers +IR	10.12	**1.465**	73.00	0.694
Latent Reward Registers +MH	10.48	1.413	74.10	0.723

Table 7: **Inference cost of RGS on SD3-Medium.** Comparison of trainable parameter budgets, memory footprints, and computational overhead against existing guided baselines. SD3 with standard CFG is provided as the unguided reference. ΔFLOP/step denotes the marginal computational overhead relative to a base CFG step. The *early-only* configuration applies guidance exclusively during the first 18 of 42 steps (where $\tau_i > 0.8$). **Bold** marks the best efficiency among guided methods (lower is better).

Method	Params↓ (M)	Peak mem↓ (GiB)	Reward Eval./step↓ (ms)	ΔFLOP/step↓ (TFLOP)	Latency/img↓ (s, ×CFG)
Reference (no reward)					
SD3 + CFG	—	16.9	—	12.4 (base)	2.99 (1.00×)
Guided baselines (separate reward backbone)					
PAVRM	691.4	31.3	**40**	+**4.1**	4.98 (1.67×)
Diffusion Probe	**15.3**	34.3	108	+11.5	7.85 (2.63×)
Decode+ImageReward	446.7	27.3	215	+10.2	12.32 (4.12×)
Latent Reward Registers (frozen generator, reward register)					
Latent Reward Registers +HPS	147.6	**17.6**	50	+5.9	5.38 (1.80×)
Latent Reward Registers +HPS *(early-only)*	147.6	**17.6**	50	+5.9	**4.00** (**1.34**×)

End-to-end latency is $1.80\times$ that of standard CFG, slightly above PAVRM ($1.67\times$) but far below Diffusion Probe ($2.63\times$) and the decode-then-ImageReward pipeline ($4.12\times$). Restricting guidance to the early phase ($\tau_i > 0.8$, i.e., 18 of the 42 steps) lowers the overhead to $1.34\times$ CFG while keeping roughly 80% of the HPSv3 gains of the full scheduler.

E.4 RGS across Sampling Budgets and Strengths

Fig. 12 evaluates uniform guidance strengths under sampling budgets of 10, 20, 30, and 40 steps, in contrast to the three-band scheduler used in the main deployment (42 steps). Guidance improves every metric relative to the CFG baseline at all budgets, and the targeted reward metrics generally rise with guidance strength. No-reference metrics such as CLIP-IQA and MUSIQ also exceed the CFG baseline but plateau or vary only marginally at higher strengths. The parameter λ in the figure legends is mathematically equivalent to the guidance strength α defined in the method section.

E.5 Reward along the RGS Trajectory

The reward-gradient correction exhibits progressive accumulation along the generation trajectory, proportionally scaling with the specified guidance magnitude. This auxiliary analysis uses a uniform guidance strength $\alpha \in [0.10, 0.20]$ rather than the three-band scheduler of the main deployment. Fig. 11 tracks the register-head scores for HPSv3, PickScore, and ImageReward through the denoising process. The correction raises the scores by 2–3.3 standard deviations relative to the CFG baseline mean, and the ordering of guidance strengths is preserved across all three evaluators.

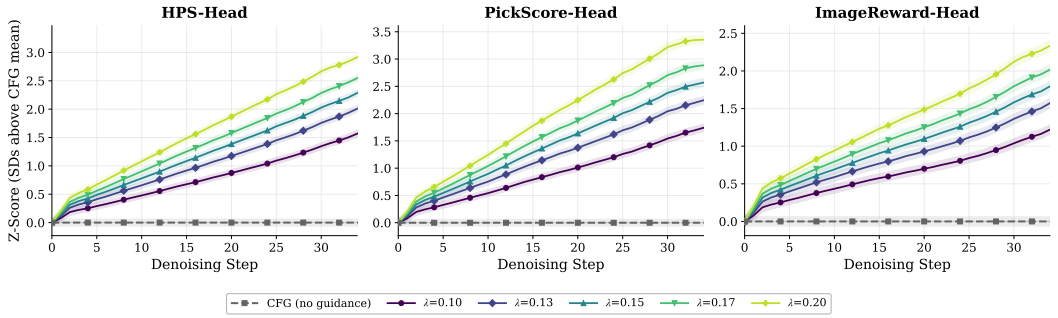

Fig. 11: **Reward progression during generation under uniform guidance.** Register-head evaluations are plotted as z-scores, normalized against the per-step distribution of the unguided CFG baseline. Curves denote different guidance strengths α (labeled as λ in the legend), with shaded regions indicating variance across diverse text prompts.

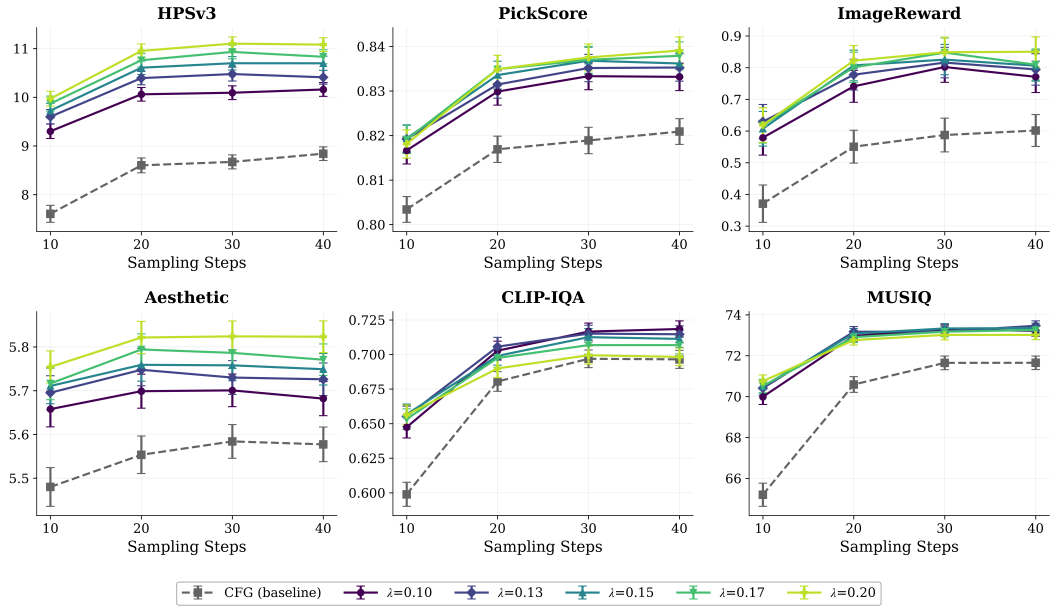

Fig. 12: **Effect of sampling budgets and guidance strengths.** Each panel plots one metric against the total number of generation steps, comparing the CFG baseline with RGS under uniform guidance strengths α (labeled as λ). Target rewards increase with guidance strength, while no-reference perceptual quality remains above the CFG baseline and plateaus at higher strengths.

E.6 Robustness to the ODE Solver

The primary results in the main text are based on first-order Euler integration of the generative flow ODE (Lipman et al., 2023). To verify whether RGS generalizes beyond this specific discretization, we evaluate the RGS protocol detailed in Appendix F.6 using four additional classical numerical integrators (Hairer et al., 1993). These include the two-step Adams–Bashforth (AB2) method, which is a multistep method requiring a single model evaluation per step analogous to Euler; the explicit midpoint method; Heun's method (a trapezoidal predictor–corrector); and the classical fourth-order Runge–Kutta (RK4) method. RGS composes with higher-order solvers in the same way as with Euler: the reward gradient is computed once at the beginning of each step, the solver's internal velocity evaluations use standard CFG forwards, and the reward-gradient correction from Sec. 3.4 is applied to the solver output afterwards. Its magnitude is scaled relative to the step displacement, $\alpha_i \cdot \mathrm{RMS}(\Delta_{\mathrm{ref},i})$, with no hyperparameter re-tuning. For these experiments, we use the same three-band

Table 8: **Performance of RGS across ODE integrators.** For the CFG rows, FID↓ is computed relative to the unguided Euler CFG reference set (integrator-induced drift); for the RGS rows, relative to the *corresponding same-solver* CFG reference set (drift attributable to RGS). NFE denotes the number of function evaluations per step, α_{mid} the mid-band guidance strength, and IR ImageReward. RGS improves target reward and perceptual quality metrics under every solver.

Solver	**Method**	**Reward Model ↑**			**Quality / Mode drift**		
(NFE)	(α_{mid})	**HPSv3**	**IR**	**PickScore**	**FID↓**	**MUSIQ↑**	**CLIP-IQA↑**
Euler (1)	CFG	8.33	0.977	0.870	0.0	73.0	0.697
	+Latent Reward Registers (0.05)	10.34	1.167	0.888	64.8	74.9	0.730
	+Latent Reward Registers (0.10)	10.44	1.154	0.888	69.3	74.2	0.712
AB2 (1)	CFG	8.29	0.995	0.869	40.1	72.8	0.686
	+Latent Reward Registers (0.05)	10.31	1.167	0.887	66.5	74.8	0.731
	+Latent Reward Registers (0.10)	10.37	1.160	0.887	72.2	74.2	0.716
Midpoint (2)	CFG	8.35	0.996	0.870	45.0	72.7	0.681
	+Latent Reward Registers (0.05)	10.14	1.167	0.887	63.4	74.7	0.728
	+Latent Reward Registers (0.10)	10.25	1.155	0.886	69.1	74.2	0.715
Heun (2)	CFG	8.27	1.008	0.869	42.6	72.8	0.687
	+Latent Reward Registers (0.05)	10.11	1.167	0.887	64.2	74.6	0.727
	+Latent Reward Registers (0.10)	10.25	1.155	0.886	69.7	74.1	0.716
RK4 (4)	CFG	8.38	1.006	0.871	46.8	72.9	0.685
	+Latent Reward Registers (0.05)	10.17	1.170	0.888	63.8	74.6	0.728
	+Latent Reward Registers (0.10)	10.27	1.152	0.887	70.0	74.0	0.715

RGS schedule. The guidance strengths are set to α_{early}=0.30 for the high-noise phase (τ_i>0.8) and $\alpha_{\text{mid}} \in \{0.05, 0.10\}$ for the mid-range phase (τ_i>0.2). The latter encompasses both the standard deployed configuration and an augmented mid-band variant, while guidance is omitted in the low-noise tail. The evaluation is conducted on a subset of 250 prompts from the balanced benchmark, with two seeds per prompt. Generations are performed at 1024^2 resolution over 42 steps with a CFG scale of 4.5 using single-head RGS. Each solver's respective CFG baseline and the two RGS configurations are evaluated on matched prompt–seed pairs. Because these runs use a 250-prompt subset, the baseline CFG numbers deviate slightly from those in Table 3, which uses the full 400-prompt set.

The findings are summarized as follows:

(1) RGS is robust across solvers. As Table 8 shows, RGS reproduces the gains seen with Euler under every integrator: paired per-image HPSv3 rises by roughly two points over the corresponding same-solver CFG baseline, with consistent improvements in the other optimized reward metrics and in no-reference quality. The drift attributable to RGS stays controlled relative to each solver's baseline, so the magnitude-matched correction generalizes across integrators without per-solver tuning or an observed compromise in no-reference quality.

(2) Cross-solver FID mostly reflects trajectory decorrelation, not generative degradation. Changing the integrator introduces an FID of 40–47 relative to the Euler CFG baseline while the other five metrics stay similar. For a fixed latent seed, different solvers converge to distinct but qualitatively comparable endpoints, so the drift induced by guidance must be measured against the corresponding solver's own baseline, as in Table 8.

(3) Higher-order solvers reduce the reward gain only marginally. Midpoint, Heun, and RK4 yield HPSv3 gains of +1.79 to +1.98, versus +2.01 to +2.11 for Euler and AB2. We hypothesize that applying the reward displacement at the end of each step exposes it to the internal evaluations of higher-order solvers, which partially undo the off-manifold correction during the next integration step. The reduction is approximately 6–11% relative to the Euler gain and does not change the qualitative conclusions. Since the single-evaluation AB2 solver closely matches Euler, both offer the best reward–quality trade-off among the integrators tested.

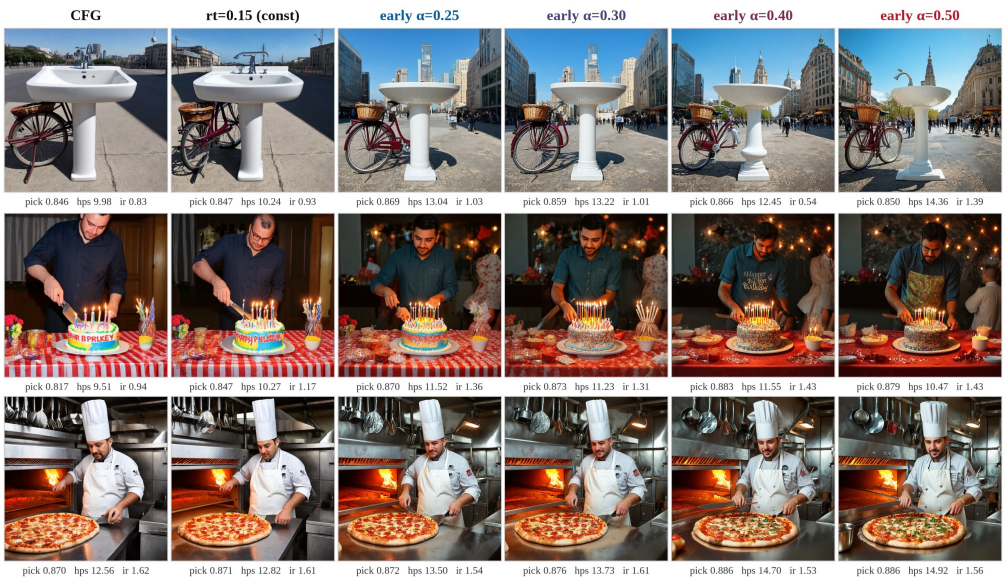

Fig. 13: **Effect of guidance strength on generation.** Samples generated under monotonically increasing guidance strengths α with a fixed prompt and initial latent. Increasing α progressively strengthens preference-aligned details.

E.7 Qualitative Comparison Across Prompts

We additionally test RGS under a flat strength α=0.15, complementing the deployed schedule in Table 10. Fig. 13 uses a matched prompt and seed to isolate the effect of guidance strength. Increasing α progressively strengthens preference-aligned details.

F Experimental Settings

F.1 Reward Register Implementation

We initialize K learnable, position-free register tokens $Q^{(0)} \in \mathbb{R}^{K\times d_h}$ as global readouts, where d_h is the hidden width of the frozen DiT. For each latent state, we set $Q_t^{(0)} = Q^{(0)}$. Let $H_t^{(\ell)} = [H_{t,\mathrm{img}}^{(\ell)}, H_{t,\mathrm{txt}}^{(\ell)}]$ denote the native image and text hidden states at block ℓ for latent z_t. For the initial L transformer blocks, the registers share a single trainable query projection W_Q^{ϕ} while reusing the frozen key, value, output, normalization, and time-conditioning modules from each block. Letting $\bar{Q}_t^{(\ell)}$, $\bar{H}_{t,\mathrm{img}}^{(\ell)}$, and $\bar{H}_{t,\mathrm{txt}}^{(\ell)}$ denote the t-conditioned normalized states, the update is:

$$\begin{aligned}
\mathbf{K}_t^{(\ell)} &= \left[\bar{Q}_t^{(\ell)} W_K^{(\ell)},\ \bar{H}_{t,\mathrm{img}}^{(\ell)} W_K^{(\ell)},\ \bar{H}_{t,\mathrm{txt}}^{(\ell)} W_{K,\mathrm{txt}}^{(\ell)}\right], \\
\mathbf{V}_t^{(\ell)} &= \left[\bar{Q}_t^{(\ell)} W_V^{(\ell)},\ \bar{H}_{t,\mathrm{img}}^{(\ell)} W_V^{(\ell)},\ \bar{H}_{t,\mathrm{txt}}^{(\ell)} W_{V,\mathrm{txt}}^{(\ell)}\right], \\
Q_t^{(\ell+1)} &= Q_t^{(\ell)} + \gamma_t^{(\ell)} \odot \mathrm{Attn}\Big(\bar{Q}_t^{(\ell)} W_Q^{\phi},\ \mathbf{K}_t^{(\ell)},\ \mathbf{V}_t^{(\ell)}\Big)\, W_O^{(\ell)},
\end{aligned} \tag{14}$$

where $[\cdot,\cdot]$ denotes concatenation along the token axis with tokens arranged as rows, $\mathrm{Attn}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathrm{softmax}(\mathbf{Q}\mathbf{K}^\top/\sqrt{d_h})\mathbf{V}$ uses bold operands for generic attention matrices to distinguish them from the register count K and register states Q, $\gamma_t^{(\ell)}$ is the frozen backbone's t-conditioned attention gate, and $\odot$ denotes element-wise multiplication. This design isolates the side stream: native image and text streams are computed independently of the registers, and for single-stream architectures (e.g., FLUX.1-dev) the native sequence serves directly as the combined input context. The registers also bypass the block feed-forward networks, so the hidden states and velocity predictions of the original DiT backbone stay strictly invariant. Finally, the register states are fused with pooled frozen features from a selected block set $\mathcal{B}$ through a lightweight attention module, producing shared per-register features $U_t \in \mathbb{R}^{K\times d_h}$ for the reward heads.

Algorithm 1: Reward-Gradient On-Policy Distillation (RG-OPD).

Input: prompt distribution $\mathcal{C}$; frozen reference field v_{ref}; frozen reward register $\{r_\phi^{(m)}\}$; schedule $\{\tau_i\}_{i=0}^{N}$; guidance strengths $\{\alpha_i\}_{i=0}^{N-1}$; selected heads $\mathcal{M}$; optimized steps $\mathcal{J}$
Output: reward-aligned student field v_θ
Initialize $v_\theta \leftarrow v_{\mathrm{ref}}$;
for *each training round* **do**
  Sample prompts $c \sim \mathcal{C}$;
  Sample $z_{\tau_0} = z_1 \sim \mathcal{N}(0, I_D)$ and roll out the current student ; `// no_grad`
  for $i \in \mathcal{J}$ **do**
    Evaluate $\mu_{\mathrm{ref},i} = \mu_{\mathrm{ref},i}(\mathrm{sg}(z_{\tau_i}), c)$ and $\Delta_{\mathrm{ref},i} = \mu_{\mathrm{ref},i} - \mathrm{sg}(z_{\tau_i})$;
    if $\alpha_i > 0$ **then**
      Compute and detach $\widehat{g}_i^{\mathcal{M}}$ using Eq. 6;
    else
      Set $\widehat{g}_i^{\mathcal{M}} \leftarrow 0$;
    end
    Construct and store $\mathrm{sg}(\mu_{\mathrm{tar},i})$ using Eq. 8;
  end
  for $i \in \mathcal{J}$ **do**
    Evaluate $\mu_{\theta,i}(\mathrm{sg}(z_{\tau_i}), c)$ and accumulate the loss in Eq. 9;
  end
  Perform a single optimizer step on the accumulated loss, updating only θ;
end

Algorithm 2: Reward-Guided Sampling.

Input: prompt c; frozen reference field v_{ref}; frozen reward register $\{r_\phi^{(m)}\}$; schedule $\{\tau_i\}_{i=0}^{N}$; guidance strengths $\{\alpha_i\}_{i=0}^{N-1}$; selected heads $\mathcal{M}$
Output: generated image x
Sample $z_{\tau_0} = z_1 \sim \mathcal{N}(0, I_D)$;
for $i \leftarrow 0$ **to** $N-1$ **do**
  Evaluate $\mu_{\mathrm{ref},i} = \mu_{\mathrm{ref},i}(z_{\tau_i}, c)$ and $\Delta_{\mathrm{ref},i} = \mu_{\mathrm{ref},i} - z_{\tau_i}$;
  if $\alpha_i > 0$ **then**
    Compute $\widehat{g}_i^{\mathcal{M}}$ using Eq. 6;
    Set $z_{\tau_{i+1}}$ using the magnitude-matched rule ; `// Eq. 8`
  else
    Set $z_{\tau_{i+1}} \leftarrow \mu_{\mathrm{ref},i}$;
  end
end
return $x = \mathrm{Dec}(z_{\tau_N})$;

F.2 Extension to Higher-Order Solvers

For a higher-order deterministic integrator, $\mu_{\mathrm{ref},i}$ is replaced by that integrator's output over $[\tau_i, \tau_{i+1}]$. The correction remains a single post-step update scaled by the resulting displacement.

F.3 Complete Algorithms

Algorithms 1 and 2 provide the complete procedures for training-time distillation and inference-time guidance, respectively.

F.4 Experimental Protocols

For latent reward prediction, we prepend $K{=}32$ register tokens to a frozen SD3-Medium backbone and train only the auxiliary branch. We measure pairwise accuracy on $54{,}170$ held-out pairs from

ImageReward (Xu et al., 2023), HPDv2 (Wu et al., 2023), HPDv3 (Ma et al., 2025), and GenAI-Bench (Li et al., 2024). Latent reward models score pairs forward-noised to $t{=}0.8$, while endpoint models score clean decoded images. Baselines include LRM (Zhang et al., 2025), PAVRM (Mi et al., 2025), Diffusion Probe (Cui et al., 2026), and DiNa-LRM (Liu et al., 2026); the FLUX.1-dev transfer results are reported in Appendix C.

Training-efficiency curves use 256 prompts, whereas Table 2 reports the full 400/800 prompt–seed evaluation. For RG-OPD, we compare against ReFL, AlignProp, Flow-GRPO, and Diffusion-NFT. All methods update generator weights and use matched prompts, seeds, resolutions, and metric pipelines; HPSv3 and ImageReward are optimization targets, while MUSIQ and CLIP-IQA measure reward-independent quality. For RGS, we compare against frozen-generator baselines DNO and Demon under matched prompts, seeds, resolutions, and metrics. DNO and Demon use external reward models for per-sample search or optimization, whereas RGS uses register gradients. HPSv3 and ImageReward measure target alignment, and MUSIQ (Ke et al., 2021) and CLIP-IQA (Wang et al., 2023) measure perceptual quality. Extended protocols and additional analyses are provided below and in Appendices E and C.

F.5 Reward Register Training

Table 9 summarizes the backbone-specific architecture and shared optimization settings for reward-register training.

Table 9: **Reward-register training details.** Backbone-specific architectural choices and shared optimization hyperparameters. Values spanning both columns are shared across architectures.

Setting	**SD3-Medium**	**FLUX.1-dev**
Register-equipped blocks L	12 (of 24)	28 (of 57; 19 double + 38 single)
Feature-snapshot blocks $\mathcal{B}$	$\{4, 8, 12\}$	$\{9, 19, 28\}$
Register tokens K	32	
Image-token pooling	64×64 to 16×16	
Readout attention heads	8	
Endpoint reward objectives	HPSv3, ImageReward	
Training data	30,652 prompt groups, 4 images/group	
Validation data	300 prompt groups	
Data split	$98/1/1$ by prompt	
Training length	3 epochs	
Batch size	64	
Precision / optimizer	bf16 AdamW	
Learning rate	5×10^{-5}	
Weight decay	0.01	
Warmup / schedule	1,000 steps / constant	
Gradient clipping	1.0	
EMA decay	0.999	

F.6 Reward-Guided Sampling

Table 10 reports the sampling configurations and guidance schedules for both backbones.

F.7 Reward-Gradient On-Policy Distillation Setting

Table 11 lists the optimization configurations and hyperparameters for SD3-Medium and FLUX.1-dev. Direct epoch-to-epoch comparisons between RG-OPD and the online RL baselines would be misleading, since the two paradigms consume data at different rates. The main text therefore reports computational efficiency in aggregate GPU hours.

Table 10: **Reward-Guided Sampling details.** Both SD3-Medium and FLUX.1-dev employ identical flow-time bands, with backbone-specific sampling step counts and mid-band guidance magnitudes.

Setting	SD3-Medium	FLUX.1-dev
Resolution	1024×1024	1024×1024
Solver	FlowMatch Euler	FlowMatch Euler
Sampling steps	42	40
Text guidance	CFG scale 4.5	Embedded guidance 3.5
Early band ($\tau_i > 0.8$)	$\alpha_{\text{early}} = 0.30$	$\alpha_{\text{early}} = 0.30$
Mid band ($0.2 < \tau_i \leq 0.8$)	$\alpha_{\text{mid}} = 0.05$	$\alpha_{\text{mid}} = 0.10$
Low-noise tail ($\tau_i \leq 0.2$)	0	0
Multi-head combination	Equal-weight sum of unit-RMS per-head gradients	
Correction mode	Additive; no spatial gradient filtering	

Table 11: **RG-OPD optimization details.** Model checkpoints were selected using a validation set disjoint from the final test set. Computational budgets are reported in terms of GPU hours and optimizer updates to facilitate fair comparison across paradigms with varying rollout densities.

Setting	SD3-Medium Configuration	FLUX.1-dev Configuration
Student adaptation	LoRA; merged for evaluation	LoRA; merged for evaluation
Training length	60–100 epochs	150 epochs
Training rollout	10 FlowMatch Euler steps	10 FlowMatch Euler steps
Optimized denoising steps $\mathcal{J}$	First 9 of 10 rollout steps ($\rho = 0.99$)	First 9 of 10 rollout steps ($\rho = 0.99$)
Teacher reward tilt α_i	$\{0.20, 0.40, 0.50\}$ for τ_i>0.2; 0 otherwise	0.80 for τ_i>0.2; 0 otherwise
Flow-GRPO / Diffusion-NFT	LoRA r=32/α=64; 6-step SDE; 100 epochs	LoRA r=32/α=64; 6-step SDE; 100 epochs
Rollout samples (RL baselines)	1,152 rollout images per epoch	1,152 rollout images per epoch

G Future Work

The central premise of this work is that generative pretraining and model scaling may improve latent reward models. Our results show that internal DiT representations support effective reward prediction, and that reward-gradient guidance on noisy latents can improve preference alignment under a frozen generator while preserving the evaluated no-reference quality metrics. Three directions seem especially promising.

(1) Few-Step Architectures and Video Generation. A natural next step is to test whether latent reward registers transfer to few-step distilled samplers and video diffusion models. For video, the reward mechanism will need structural adaptations to enforce temporal consistency and coherent motion.

(2) From RLHF to Reinforcement Learning from Verifiable Rewards (RLVR). The framework is not limited to human-preference optimization. We plan to explore its integration with RLVR, where the same latent readout could optimize objective criteria such as factual fidelity, spatial reasoning, and physical plausibility.

(3) Scaling DiT Reward Models. A promising direction is to scale both the register stream and its reward readout. Increasing the number of registers and grouping them could capture complementary aspects of reward. Beyond the current MLP heads, a vision–language model conditioned on register features, text, and references could evaluate difficult criteria such as fine-grained text rendering, action semantics, temporal coherence, and identity consistency. Studying how performance scales with register count, evaluator capacity, and supervision would clarify the trade-off between richer reward signals and guidance cost.