SELF-OPD: ON-POLICY DISTILLATION FOR FLOW MATCHING MODELS WITHOUT TEACHER

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation (OPD) leverages a pre-trained, specialized teacher model to provide dense supervisory signals. This method has achieved significant success in Large Language Models (LLMs) and has recently been adapted to flow matching models. However, this paradigm suffers from two major issues: First, training a separate, task-specific teacher for every new objective incurs prohibitive computational and time costs. Second, task-conditioned integration of multiple specialized teachers does not explicitly require task fidelity and preference quality to coexist in each generated image, and can leave output quality dependent on the prompt distribution. In this paper, we introduce **Self-OPD**, a teacher-free OPD framework for flow matching models that turns the student's own self-exploration into step-wise supervision. At each timestep, Self-OPD branches the deterministic next-state prediction into K stochastic SDE candidates, rolls them out with the ODE sampler, and compares their rewards against a deterministic self-reference baseline to obtain normalized advantages. The velocity field is optimized with an all-branch pull-push objective, where positive-advantage branches pull the student toward better trajectories, while negative-advantage branches push it away, regulated by direction-aware attenuation and SDE-variance normalization. For multi-objective alignment, Self-OPD brings reward-level fusion back into OPD, jointly scoring each sampled image before trajectory distillation so that task fidelity and preference quality can be optimized within the same output. Experiments on single- and multi-reward benchmarks show that Self-OPD outperforms prior RL and OPD methods without task-specific teachers.

1 INTRODUCTION

Flow Matching (FM) models Lipman et al. (2022); Liu et al. (2022); Esser et al. (2024), which learn a continuous velocity field that transports noise to data, have become a dominant backbone for high-quality visual generation Ho et al. (2020); Rombach et al. (2022); Podell et al. (2024); Black Forest Labs (2024). Despite their generative strength, aligning FM models with downstream objectives such as text rendering Chen et al. (2023); Yang et al. (2023), compositional correctness, and human preference remains challenging.

Existing reinforcement learning (RL) approaches Liu et al. (2026); Zheng et al. (2026); Black et al. (2024); Fan et al. (2023); Wallace et al. (2024); Wang et al. (2026) directly optimize reward signals by treating generation as a sequential decision process. As summarized in the top-left panel of Fig. 1 and illustrated in its bottom panel, these methods can be teacher-free, but they often suffer from challenging trajectory-level credit assignment. Because rewards are only evaluated at the final step, assigning credit backward through numerous denoising steps leads to high-variance gradients and makes multi-objective alignment fragile when different rewards favor different visual properties.

On-policy distillation (OPD) provides a more stable alternative by supplying dense, step-wise supervision on the student's own trajectory. Flow-OPD Fang et al. (2026) establishes the teacher-based formulation for flow matching models by matching the student's velocity or transition target to a pre-trained reward-specialized teacher at each denoising step. DiffusionOPD Li et al. (2026) extends this paradigm to multiple objectives: it trains task-specific teachers independently and integrates their capabilities into a unified student by matching each task dataset to its corresponding teacher field. Both methods therefore distill supervision at the velocity-field or transition-target level. This design

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.

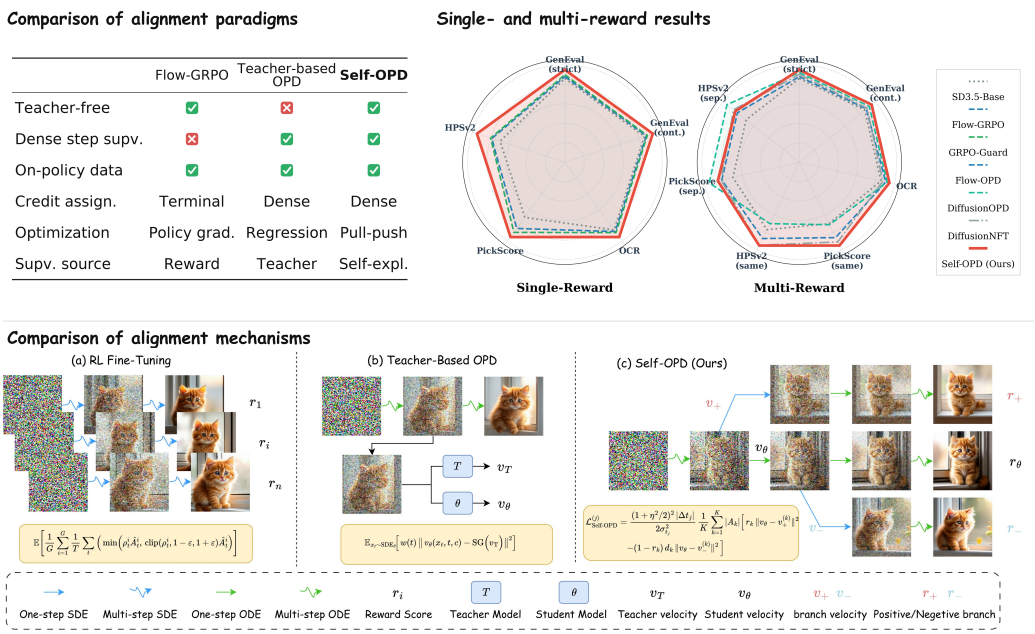

Figure 1: **Overview of Self-OPD.** *Comparison of alignment paradigms:* We compare the supervision, exploration, and teacher requirements of different methods. *Single- and multi-reward results:* Self-OPD performs strongly in both settings without task-specific teachers. *Comparison of alignment mechanisms:* (a–c) compare trajectory-level policy gradients, teacher-based step-wise distillation, and Self-OPD's self-exploration-based pull-push supervision, respectively.

requires training or acquiring a specialized teacher for each objective, leaves the student bounded by its teachers, and, in the multi-teacher case, does not explicitly require a single generated image to retain task fidelity and preference quality at the same time Fang et al. (2026); Li et al. (2026); Zhou et al. (2026). This motivates a central question: ***Can we retain the dense per-step supervision of OPD while removing the need for any external teacher?***

We answer this question with **Self-OPD**, a teacher-free on-policy distillation framework for flow matching models. The key idea is to replace teacher-provided velocity targets with reward-weighted targets discovered from the student's own local exploration. At each on-policy timestep t_j, Self-OPD branches the deterministic next-state prediction $x_{t_{j+1},\theta}$ into K stochastic SDE candidates, rolls each candidate out deterministically to a clean image, and evaluates the resulting images with task rewards. In parallel, a deterministic ODE rollout from the same parent state provides a self-reference baseline, allowing terminal rewards to be converted into normalized branch advantages. Self-OPD then performs all-branch pull-push distillation: high-advantage branches pull the velocity field toward better local trajectories, while low-advantage branches push it away from poor directions. A direction-aware attenuation term stabilizes this signed objective by suppressing repulsion that would counteract the best branch, and an SDE-variance normalization yields a principled per-step alignment loss connected to reward-tilted KL minimization. This self-referenced formulation also changes how multi-objective alignment is handled. Self-OPD evaluates each sampled image with all rewards applicable to the current task, fuses their normalized scores at the reward level, and uses the composite reward only to rank and weight sampled branches. The regression target remains a concrete trajectory velocity from the student's own neighborhood, so black-box rewards can jointly guide the same output without being differentiated. Compared with task-conditioned teacher-field integration, this construction explicitly searches for image trajectories that jointly retain task fidelity and preference quality.

Overall, our contributions are:

- We propose **Self-OPD**, a teacher-free on-policy distillation framework for flow matching models that converts reward-guided self-exploration and a self-reference baseline into dense step-wise supervision.

- We introduce an all-branch pull-push distillation objective with SDE-variance normalization and direction-aware attenuation, exploiting positive-advantage branches to pull and negative-advantage branches to push while maintaining stable optimization.
- We reintroduce reward-level fusion into OPD by jointly scoring each sampled image prior to trajectory distillation, rather than combining objectives via teacher velocity.
- We show empirically that Self-OPD outperforms prior RL and teacher-based OPD methods across single- and multi-reward benchmarks.

2 Related Work

RL in Diffusion Models. RL has recently emerged as an effective paradigm for aligning diffusion-based visual generation models Rombach et al. (2022); Podell et al. (2024); Esser et al. (2024); Black Forest Labs (2024); Liu et al. (2025a;b) with human preferences Liu et al. (2026); Zheng et al. (2026); Xue et al. (2025a); Black et al. (2024); Fan et al. (2023); Wallace et al. (2024); Clark et al. (2024); Xu et al. (2023). From the perspective of backward estimation, FlowGRPO Liu et al. (2026), TempFlow-GRPO He et al. (2026), TP-GRPO Tong et al. (2026), and DanceGRPO Xue et al. (2025b) reformulate denoising as a sequential decision-making problem, enabling policy gradient optimization over discrete diffusion timesteps. DiffusionNFT Zheng et al. (2026) instead operates on the forward diffusion trajectory, treating the noising process as an environment to derive reward signals for policy updates. More recently, works Wang et al. (2026); Zheng et al. (2026) propose multi-reward optimization to improve generation quality across multiple criteria while preserving generalization, formulating a multi-task RL framework for visual generation.

On-Policy Distillation. On-policy distillation (OPD) Agarwal et al. (2024); Fang et al. (2026); Li et al. (2026); Yuan et al. (2026); Wu et al. (2026) has demonstrated notable progress in large language models (LLMs), owing to its ability to mitigate exposure bias and improve training efficiency through student-generated sequences Agarwal et al. (2024); Singh et al. (2023). GKD Agarwal et al. (2024) established the canonical OPD framework by training the student model on its own generated outputs, effectively bridging the distribution gap between teacher and student. Building upon this, Flow-OPD Fang et al. (2026) transfers the capability of a reward-specialized teacher through velocity-field or transition-target matching, while DiffusionOPD Li et al. (2026) extends this teacher-based formulation to task-conditioned integration of multiple specialized teachers into one student. These methods extend OPD Fu et al. (2026b;a) to flow matching and diffusion models and achieve competitive generation quality with significantly reduced sampling steps Yin et al. (2024b;a). D-OPSD Jiang et al. (2026) further leverages multi-modal features to distill rich cross-modal knowledge into a compact single-stream textual representation, enhancing semantic alignment in text-to-image generation. In contrast, our work extends OPD to a multi-task setting for simultaneous optimization across diverse objectives, without a fixed teacher model.

3 Methodology

In this work, we propose **Self-OPD**, a teacher-free on-policy distillation framework that converts self-generated rewards into dense, step-wise supervision. As shown in Fig. 2, Self-OPD explores local trajectories around the student's on-policy path via stochastic SDE branching (Fig. 2a), computes self-referenced advantages against a deterministic self baseline (Fig. 2b), and distills the feedback with a gated advantage-weighted pull-push objective (Fig. 2c).

3.1 Self-Referenced Evaluation and Self-Exploration

To retain dense supervision without a pretrained teacher, Self-OPD lets the student evaluate its own local alternatives. Given an on-policy latent x_{t_j}, the method first samples K stochastic next states around the deterministic prediction and then compares their final rewards against the student's default ODE trajectory from the same state. This local, self-referenced comparison turns terminal rewards into low-variance per-step guidance.

SDE Branching Exploration. At timestep t_j, we compute the deterministic next-state prediction $x_{t_{j+1},\theta}$ with a single forward pass through the student transformer. Following the SDE transition in

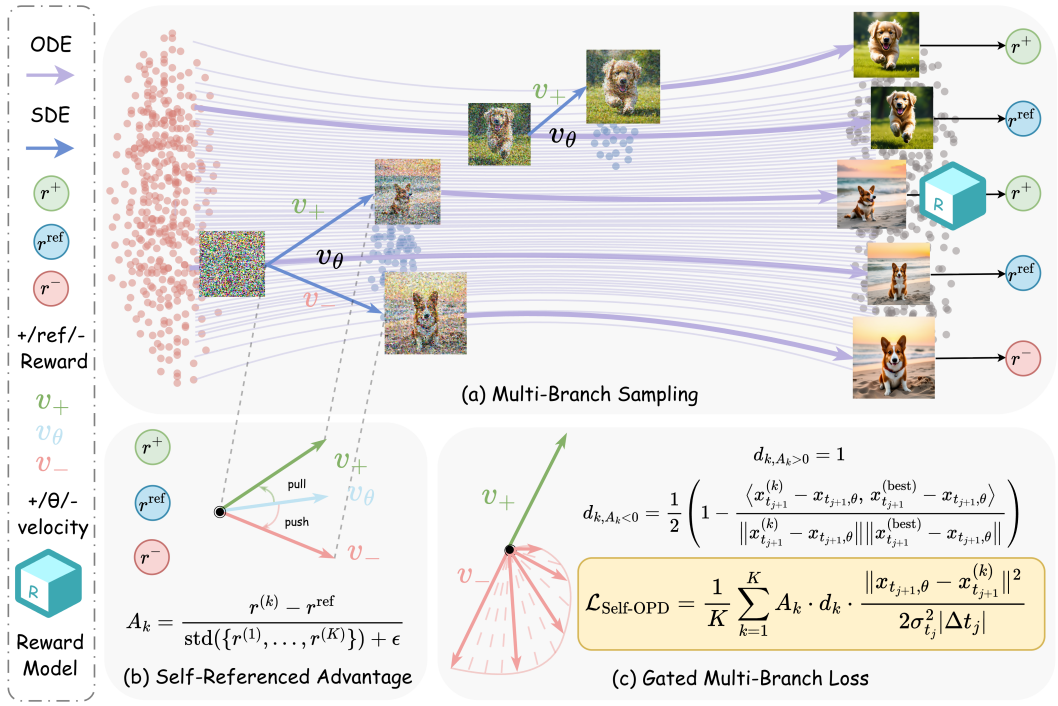

Figure 2: **Self-OPD pipeline. (a) Multi-Branch Sampling:** the student branches its prediction into K SDE paths (blue), generates images via ODE rollouts (purple), and scores them (r^+, r^{ref}, r^-). **(b) Self-Referenced Advantage:** rewards define a self-referenced advantage A_k that pulls toward high-reward velocities v_+ and pushes away from low-reward velocities v_-. **(c) Gated Multi-Branch Loss:** a direction-aware coefficient d_k gates the push to prevent it from counteracting the pull, resulting in the multi-branch loss $\mathcal{L}_{\text{Self-OPD}}$.

Eq. 11, we instantiate K candidate branches by drawing independent Gaussian perturbations:

$$x_{t_{j+1}}^{(k)} = x_{t_{j+1},\theta} + \sigma_{t_j}\sqrt{|\Delta t_j|}\, z_k, \quad z_k \sim \mathcal{N}(0, I) \tag{1}$$

while $k = 1, \ldots, K$, all branches share the same base prediction $x_{t_{j+1},\theta}$, so the transformer is evaluated only once; diversity is introduced via lightweight SDE perturbations. The perturbation scale σ_{t_j}, controlled by $\eta = 0.6$, sets the exploration radius.

ODE Rollout and Self-Reference Evaluation. Each branch $x_{t_{j+1}}^{(k)}$ is completed to a clean latent $\hat{x}_0^{(k)}$ by a deterministic ODE rollout (η=0), decoded to pixel space, and scored by a task-specific reward model R:

$$r^{(k)} = R\left(\text{Dec}(\hat{x}_0^{(k)}),\, c\right), \quad k = 1, \ldots, K, \tag{2}$$

where c is the prompt and $\text{Dec}(\cdot)$ is the VAE decoder. To obtain a variance-reducing baseline without a teacher, we also run a fully deterministic ODE trajectory directly from the parent state x_{t_j} and denote its reward as r^{ode}. The branch advantage is then normalized relative to this self-reference:

$$A_k = \frac{r^{(k)} - r^{\text{ode}}}{\text{std}(\{r^{(1)}, \ldots, r^{(K)}\}) + \epsilon}, \tag{3}$$

where positive advantages indicate local directions that outperform the student's default trajectory, while negative advantages mark directions to avoid. For multi-reward training, $r^{(k)}$ is the composite reward in Eq. 9; it is used only as a non-differentiable branch-ranking signal in Eq. 3.

3.2 All-Branch Pull-Push Distillation

Instead of regressing only to the best branch, Self-OPD uses the full exploration neighborhood. Positive-advantage branches should pull the deterministic prediction toward locally better trajectories, while negative-advantage branches should push it away from poor directions. This all-branch

update reduces target switching and preserves the information contained in branches that Best-of-K selection would discard.

Nevertheless, the effectiveness of this repulsive push depends heavily on the geometric alignment of the branches. If a negative-reward branch points in nearly the same direction as the best branch, a strong repulsive update would counteract the desirable pull. We therefore introduce a direction-aware attenuation coefficient:

$$d_k = \begin{cases} 1 & \text{if } A_k \geq 0, \\ \dfrac{1}{2}\left(1 - \dfrac{\langle \delta_k, \delta_{\text{best}} \rangle}{\|\delta_k\| \|\delta_{\text{best}}\|}\right) & \text{if } A_k < 0, \end{cases} \tag{4}$$

where $\delta_k = x^{(k)}_{t_{j+1}} - x_{t_{j+1},\theta}$ and $\delta_{\text{best}} = x^{(\text{best})}_{t_{j+1}} - x_{t_{j+1},\theta}$ are measured from the deterministic prediction. The coefficient is bounded in $[0, 1]$: it keeps the full repulsion for negative branches that point opposite to the best direction, and suppresses repulsion when a negative branch is aligned with the best branch.

For each sampled transition $x^{(k)}_{t_{j+1}}$, let $v^{(k)}$ denote its equivalent velocity-space target, with $v^{(k)}_+$ and $v^{(k)}_-$ denoting targets from positive- and negative-advantage branches, respectively, and let $r_k = \mathbf{1}[A_k \geq 0]$. Appendix A.2 derives the closed-form transition-to-velocity conversion and the resulting variance-dependent prefactor. The step-wise Self-OPD objective is:

$$\mathcal{L}^{(j)}_{\text{Self-OPD}} = \frac{(1 + \eta^2/2)^2 \, |\Delta t_j|}{2\sigma^2_{t_j}} \, \frac{1}{K} \sum_{k=1}^{K} |A_k| \Big[r_k \, \|v_\theta - v^{(k)}_+\|^2 - (1 - r_k) \, d_k \, \|v_\theta - v^{(k)}_-\|^2 \Big], \tag{5}$$

The prefactor follows from the SDE transition variance and the affine change of variables from transition mean to velocity, matching the normalization used in OPD-style per-step regression Fang et al. (2026); Li et al. (2026). The first term pulls v_θ toward high-reward branch velocities, while the second term pushes it away from low-reward velocities after direction-aware attenuation.

Connection to Per-Step KL. The normalization above also admits a KL interpretation. Let $q_\theta(\cdot \mid x_{t_j}) = \mathcal{N}(x_{t_{j+1},\theta}, \sigma^2_{t_j}|\Delta t_j| I)$ denote the SDE transition kernel, and let $q^*(\cdot \mid x_{t_j}) \propto q_\theta(\cdot \mid x_{t_j}) \exp(A(\cdot)/\tau)$ be a reward-tilted target distribution, as in reward-weighted regression.

Proposition 1 (Per-step KL gradient)**.** *For fixed x_{t_j} and fixed target q^*, the gradient of the reverse KL with respect to the student transition mean is*

$$\nabla_{x_{t_{j+1},\theta}} D_{\text{KL}}(q^* \| q_\theta) = \frac{1}{\sigma^2_{t_j}|\Delta t_j|} \mathbb{E}_{x_{t_{j+1}} \sim q^*} \left[x_{t_{j+1},\theta} - x_{t_{j+1}} \right]. \tag{6}$$

This identity motivates the transition-precision normalization $1/(\sigma^2_{t_j}|\Delta t_j|)$ in Eq. 5. Self-OPD then uses the K sampled branches to construct a practical signed pull–push surrogate weighted by A_k and d_k; the proof and its detailed connection to the training loss are given in Appendix B.

To aggregate step-wise losses across the denoising trajectory, we define the training objective as:

$$\mathcal{L}_{\text{total}} = \sum_{j=0}^{S-1} \alpha_j \, \mathcal{L}^{(j)}_{\text{Self-OPD}}, \tag{7}$$

where $\mathcal{L}^{(j)}_{\text{Self-OPD}}$ is the step-wise loss at timestep t_j (Eq. 5), and α_j is a timestep-dependent weight. Following the intuition that early-to-mid timesteps (closer to noise) establish the global semantic layout while late timesteps (closer to data) merely refine local details, we set α_j to prioritize these early exploration phases and smoothly decay it as the trajectory approaches convergence at $t = 0$.

3.3 Reward-Level Fusion for Multi-Objective Alignment

Task-wise capability decoupling can be effective for LLMs, where different instructions may invoke different skills. Image generation places a stronger requirement on multi-objective alignment: each generated image should jointly retain task fidelity, aesthetic quality, and human preference, rather

than achieving these scores separately on different prompt sets. Earlier multi-reward RL methods combine objectives at the reward level Zheng et al. (2026); Wang et al. (2026). In contrast, Flow-OPD Fang et al. (2026) and DiffusionOPD Li et al. (2026) both perform distillation at the velocity-field or transition-target level. Flow-OPD matches the student to a pretrained reward-specialized teacher along the student's on-policy trajectory, while DiffusionOPD extends this teacher-matching paradigm to multiple objectives by training a specialized teacher v_{ϕ_m} for each task m and matching prompts from task dataset $\mathcal{D}_m$ to the corresponding teacher. Taking DiffusionOPD as an example, its multi-task distillation objective can be written schematically in velocity form as

$$\mathcal{L}_{\text{DiffusionOPD}} = \mathbb{E}_{m,\, c\sim\mathcal{D}_m} \sum_j w_j \left\| v_\theta(x_{t_j}, t_j, c) - v_{\phi_m}(x_{t_j}, t_j, c) \right\|_2^2 . \tag{8}$$

Here m jointly indexes the current task dataset and its specialized teacher. DiffusionOPD cycles over tasks and accumulates their losses before updating the shared student. It therefore integrates multiple teacher capabilities through task-conditioned velocity or transition matching, rather than summing multiple teacher velocities for the same prompt. Because each generated image receives supervision from the teacher selected by its task, the objective does not directly require that image to be strong under the other objectives. This permits output distributions to remain decoupled across prompt families, with preference quality tied to the prompt family. As shown in Fig. 4(e,f), DiffusionOPD exhibits larger PickScore and HPSv2 gaps between aesthetic prompts and GenEval/OCR prompts than Self-OPD, even under the aesthetic co-teacher control.

Self-OPD instead returns fusion to the *reward level*. For each candidate branch k, all reward functions applicable to the current task evaluate the same decoded image. We standardize each score and form a composite reward

$$r^{(k)} = \sum_{m=1}^{M} \lambda_m \, \tilde{r}_m^{(k)}, \qquad \tilde{r}_m^{(k)} = \frac{r_m^{(k)} - \mu_m}{\sigma_m + \epsilon}, \tag{9}$$

where λ_m controls the contribution of reward m. This score is not differentiated. It is used only to compute the branch advantage in Eq. 3 and thereby rank and weight concrete trajectory targets in Eq. 5. Thus, task correctness and preference quality are evaluated together on each image before its trajectory supplies supervision.

Self-OPD brings the established reward-level fusion strategy back into OPD. Rather than integrating objectives through task-conditioned teacher velocities, it first evaluates the same sampled image under multiple rewards and then turns the jointly ranked and weighted trajectories into dense step-wise supervision. This encourages task fidelity and preference quality to coexist within individual outputs. The smaller cross-dataset preference gaps in Fig. 4(e,f) support this behavior. The reward-tilted derivation and cross-dataset consistency definition are provided in Appendix C; the practical task-wise reward configuration and normalization are described in Sec. 4.1.

4 Experiments

4.1 Experimental Settings

Implementation Details. We use SD3.5-Medium Esser et al. (2024) at 512×512 resolution as the base model. We train with LoRA Hu et al. (2021) applied to the transformer. We use $K = 8$ branches, noise level $\eta = 0.7$, and training timesteps per step $= 2$. We use the AdamW Kingma & Ba (2014) optimizer with learning rate 3×10^{-4}.

Tasks and Rewards. We evaluate three tasks with corresponding reward models: (1) *text rendering*, scored by OCR character-level accuracy between the generated and target text; (2) *compositional generation*, scored by GenEval Ghosh et al. (2023); and (3) *aesthetic and human-preference alignment*, scored by PickScore Kirstain et al. (2023) and HPSv2 Wu et al. (2023). We also train a multi-reward model that combines the task scorers.

Multi-Reward Fusion Setup. OCR and GenEval are prompt-dependent tasks and cannot both be meaningfully evaluated on the same prompt. To preserve the original training prompts used by the baselines, we keep the two datasets unchanged. During training, the same multi-reward model is updated separately on OCR-prompt batches and GenEval-prompt batches. OCR batches combine

Table 1: **Main results of single-reward training.** Best in **bold**, second best underlined among trained methods.

Method		GenEval		OCR	PickScore	HPSv2
		strict	cont.			
Base models (no alignment fine-tuning)						
SD3.5-Medium (base)		0.5222	0.6219	0.5833	22.41	0.3004
RL fine-tuning (teacher-free)						
Flow-GRPO	+ RM GenEval	0.9005	0.9470	0.6519	22.53	0.2792
	+ RM OCR	0.5420	0.6529	0.9253	22.51	0.2999
	+ RM PickScore	0.5081	0.5293	0.7033	23.57	0.3396
GRPO-Guard	+ RM GenEval	0.9155	0.9502	0.7032	22.20	0.2586
	+ RM OCR	0.5402	0.6870	0.9348	22.44	0.2912
	+ RM PickScore	0.4602	0.4037	0.6563	23.98	0.3453
Ours (teacher-free)						
Self-OPD	+ RM GenEval	**0.9536**	**0.9676**	0.6120	22.46	0.2741
	+ RM OCR	0.4186	0.5565	**0.9745**	22.11	0.2719
	+ RM PickScore	0.5303	0.5863	0.7416	**24.47**	0.3357
	+ RM HPSv2	0.4453	0.4722	0.7121	23.25	**0.4099**

Table 2: **Main results of multi-reward training.** Each method produces a single model evaluated across all metrics.

Method	Teacher-free	GenEval		OCR	Pref. (same test images)		Pref. (separate test set)	
		strict	cont.		PickScore	HPSv2	PickScore	HPSv2
Base models (no alignment fine-tuning)								
SD3.5-Medium (base)	—	0.5222	0.6219	0.5833	22.66	0.2824	22.41	0.3004
Teacher-based OPD								
Flow-OPD	✗	0.8594	0.9203	0.9392	23.43	0.3042	23.13	0.3334
DiffusionOPD	✗	0.9150	0.9479	0.9464	22.72	0.2676	**23.95**	**0.3729**
Multi-reward (teacher-free)								
DiffusionNFT	✓	0.8888	0.9277	0.9229	23.67	0.3206	23.18	0.3482
Self-OPD(Ours)	✓	**0.9521**	**0.9691**	**0.9597**	**23.87**	**0.3214**	23.39	0.3415

OCR, PickScore, and HPSv2, while GenEval batches combine GenEval, PickScore, and HPSv2; in both cases, the task reward and the two preference rewards are weighted $3:1:1$. At each training step, all SDE-branch rollout images and the pure-ODE self-reference baseline are scored by the three reward models. Each scorer is z-normalized over this joint set, and their weighted sum forms the composite score used to compute the advantage in Eq. 3 and rank the branches. The resulting model is trained without teachers and evaluated across all metrics.

Baselines. We compare against three groups. *Base models*: SD3.5-Medium Esser et al. (2024). *Teacher-free RL fine-tuning*: Flow-GRPO Liu et al. (2026), GRPO-Guard Wang et al. (2026), and DiffusionNFT Zheng et al. (2026). *Teacher-based OPD*: Flow-OPD Fang et al. (2026) and DiffusionOPD Li et al. (2026).

4.2 Main Results

Quantitative Results. We evaluate all methods across two distinct setups, namely single-reward training in Tab. 1 and multi-reward training in Tab. 2. The assessment encompasses compositional generation via GenEval, text rendering via OCR, and human/AI preferences measured by PickScore and HPSv2. For GenEval, we report both the strict mode, which binary-scores generations only when all sub-requirements are met, and the continuous mode, which allows for partial credit. Similarly, preference metrics are assessed under two protocols: the same test images protocol that evaluates directly on the GenEval and OCR benchmarks, and the separate test set protocol that utilizes a held-out aesthetic benchmark.

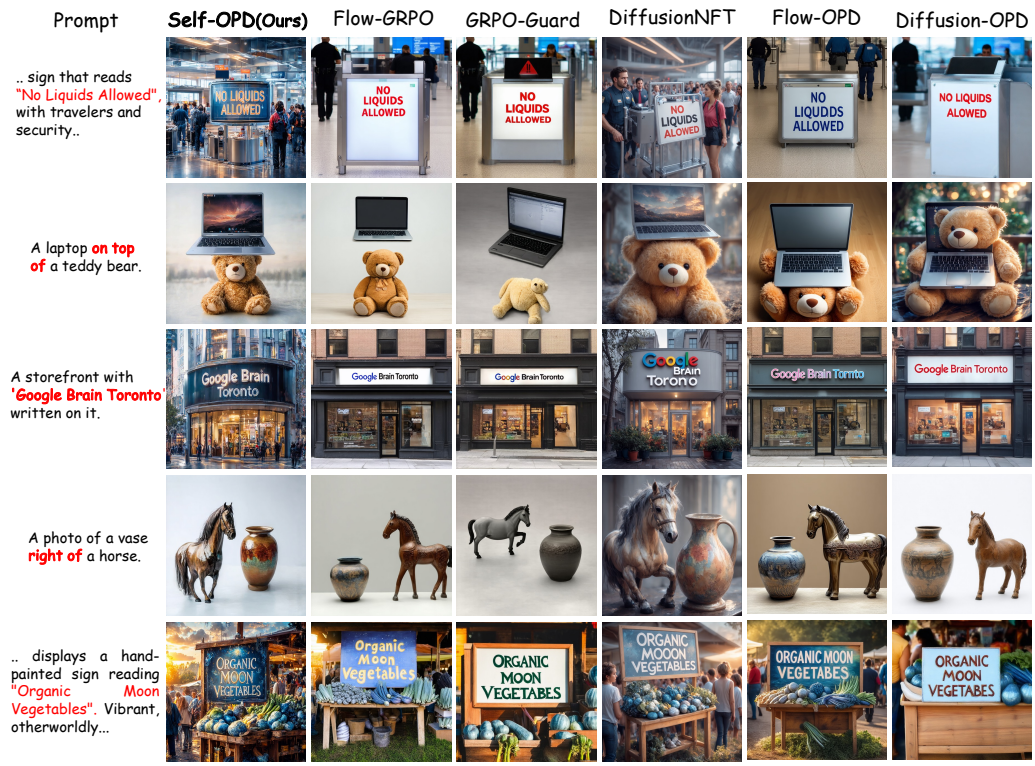

Figure 3: **Qualitative comparison.** Each method uses its multi-reward model if available, or the prompt-specific reward model (e.g., OCR for text prompts). Self-OPD achieves superior performance in both accuracy and aesthetics.

Self-OPD emerges as the sole teacher-free method that remains highly competitive across all evaluation dimensions simultaneously. In Tab. 1, among all single-reward methods, Self-OPD achieves top performance with a GenEval score of 0.95, OCR accuracy of 97.5%, PickScore of 24.79, and HPSv2 score of 0.3665.

In Tab. 2, the unified multi-reward Self-OPD model achieves the best performance on both GenEval and OCR. Under the same test images protocol, it also achieves the highest PickScore (23.87) and HPSv2 (0.3214) among all compared methods while remaining teacher-free. For image generation, we seek each image to score highly under multiple applicable rewards, rather than decoupling the objectives and achieving high scores only on separate test sets. We further examine how this result reflects the distinction between reward-level fusion and task-conditioned teacher-field integration in the cross-dataset preference consistency analysis of Sec. 4.3.

Qualitative Results. Fig. 3 compares all methods on six prompts for text rendering, spatial relations, and counting. Compared with multi-reward baselines including DiffusionNFT, Flow-OPD, and DiffusionOPD, Self-OPD produces more faithful text, better preserves complex layouts, satisfies counting constraints more reliably, and yields more coherent compositions. Compared with single-reward specialists at their optimal checkpoints, Self-OPD demonstrates stronger and more balanced control across all three aspects, improving both scene richness and aesthetic quality, consistent with the quantitative gains in Tab. 1 and Tab. 2.

4.3 Analysis and Ablations

We consolidate our analyses and ablations in Fig. 4, discussing the six panels in pairs.

Branch Selection and Repulsion Bounding. Figure 4(a) compares three branch-selection strategies. Best-of-K improves slowly and remains below the two all-branch variants. Using all branches produces more stable improvement, while adding $|\Delta t|$-aligned KL normalization achieves the best performance. These results show the benefits of using feedback from all branches and applying step-wise normalization. Figure 4(b) compares bounded and unbounded repulsion. Allowing $d_k \in [0, 2]$

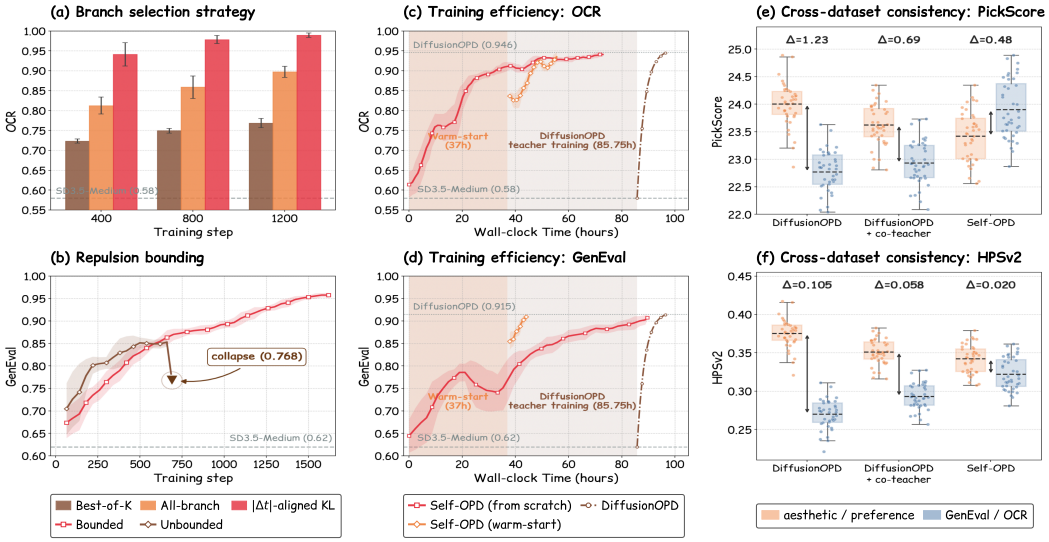

Figure 4: **Analysis of Self-OPD. (a)** All-branch pull–push distillation with $|\Delta t|$-aligned KL outperforms Best-of-K. **(b)** Bounded repulsion avoids the collapse of the unbounded variant. **(c,d)** Self-OPD reaches DiffusionOPD-level OCR and GenEval at far lower wall-clock cost. **(e,f)** Reward-level fusion produces more consistent preference scores across aesthetic and task prompts than task-conditioned teacher-field integration, including the co-teacher control.

produces faster improvement than the bounded range $[0, 1]$ during the early stage of training. However, the stronger repulsive updates also introduce instability and eventually cause training collapse. Bounding d_k to $[0, 1]$ leads to more gradual but stable improvement. This result shows that limiting the strength of repulsion is important for stable training.

Training Efficiency. For teacher-based distillation, the total training time should include both teacher training and subsequent student distillation. Fig. 4(c,d) compares the training efficiency of Self-OPD and DiffusionOPD on OCR and GenEval. According to the training times and wall-clock convention reported in DiffusionOPD (Li et al., 2026), its GenEval, OCR, and aesthetics teachers take 46.92 h, 33.17 h, and 85.75 h to train, respectively, followed by 11.26 h of student distillation. Its total wall-clock time is 97.01 h. Self-OPD trained from scratch takes approximately 90 h to reach DiffusionOPD-level performance on both tasks. Although this is longer than the 11.26 h student-distillation stage alone, it is shorter than the complete teacher-and-student training time of DiffusionOPD. With warm-start, the GenEval and OCR specialists take 27 h and 37 h to train in parallel, followed by up to approximately 11 h of multi-reward training. Its total wall-clock time is approximately 48 h, which further reduces the training time.

Cross-Dataset Preference Consistency. Fig. 4(e,f) compares PickScore and HPSv2 on aesthetic and GenEval/OCR task prompts, where a smaller gap Δ indicates greater consistency across prompt distributions. DiffusionOPD has gaps of 1.23 and 0.105, whereas Self-OPD has smaller gaps of 0.48 and 0.020, respectively. Applying an aesthetic co-teacher to the GenEval/OCR task-training sets narrows DiffusionOPD's gaps to 0.69 and 0.058, but they remain larger than Self-OPD's and the improvement comes at the cost of lower scores on aesthetic prompts. These results support that reward-level fusion maintains cross-dataset preference quality more consistently than task-conditioned teacher-field integration.

5 Conclusion

In this paper, we introduce Self-OPD, a teacher-free on-policy distillation framework for flow matching models. By branching the student's trajectory and scoring each branch with task rewards, then performing direction-aware, advantage-weighted regression over all branches, Self-OPD converts self-exploration into dense per-step supervision. Experiments on text rendering, compositional generation, and preference alignment show that our approach matches or exceeds teacher-based OPD, while avoiding task-specific teacher training and enabling reward-level multi-objective fusion.

AI USE STATEMENT

In this work, we used generative AI tools to assist with preliminary background and related-work research. We did not use generative AI tools to write the main body of the paper, create its figures or tables, construct the codebase, or run, generate, or analyze the experimental results. Generative AI output was not treated as a source of scientific evidence. We additionally used generative AI tools to identify potential errors or omissions in the manuscript and its citations. All AI-assisted suggestions were reviewed by the authors, and cited claims were checked against the original sources. We take full responsibility for the final content of this work, including its text, claims, code, figures, and other research artifacts.

ETHICS STATEMENT

This work does not involve human subjects, collect personal or private data, or release a new dataset. The experiments use publicly available pretrained models, text-prompt benchmarks, and automated evaluators. The work studies a general training method for image generation and neither targets particular demographic groups nor develops applications for high-risk domains. We are not aware of ethical concerns introduced by this work beyond those generally associated with pretrained text-to-image models. Any release or downstream use remains subject to the applicable licenses, acceptable-use terms, and standard responsible-use practices. The authors take responsibility for the conduct and presentation of this research.

REPRODUCIBILITY STATEMENT

We are committed to ensuring the reproducibility of this work. The main paper, appendix, and supplementary materials document the methodology, experimental setup, implementation details, and evaluation procedure needed to reproduce the main results. The relevant code and configurations will also be released to facilitate independent verification. Because training and generation are stochastic, minor numerical variation across runs may occur.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *ICLR*, pp. 21246–21263, 2024.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *ICLR*, pp. 4965–4987, 2024.

Black Forest Labs. FLUX.1. `https://github.com/black-forest-labs/flux`, 2024. Accessed: 2025-01-15.

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

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

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024.

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 *NeurIPS*, pp. 79858–79885, 2023.

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

Siming Fu, Zheming Fu, Ruizhe He, Hualiang Wang, Jie Huang, Xiaoxiao Ma, Mingchen Zhong, Weihu Huang, Xiaoxuan He, and Haojun Xu. Any-opd: Heterogeneous on-policy distillation for flow-matching models via representation-space bridging. *arXiv preprint arXiv:2608.03316*, 2026a.

Siming Fu, Haojun Xu, Ruizhe He, Zheming Fu, Hualiang Wang, Jie Huang, Xiaoxiao Ma, Mingchen Zhong, Weihu Huang, Xiaoxuan He, et al. Poly-opd: Heterogeneous multi-teacher on-policy distillation for capability-selectable flow models. *arXiv preprint arXiv:2608.04349*, 2026b.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. In *NeurIPS*, pp. 52132–52152, 2023.

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. In *International Conference on Learning Representations*, volume 2026, pp. 132869–132895, 2026.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

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. *arXiv preprint arXiv:2106.09685*, 2021.

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

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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 *NeurIPS*, pp. 36652–36663, 2023.

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 TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. In *NeurIPS*, pp. 40783–40818, 2026.

Mushui Liu, Yuhang Ma, Yang Zhen, Jun Dan, Yunlong Yu, Zeng Zhao, Zhipeng Hu, Bai Liu, and Changjie Fan. Llm4gen: Leveraging semantic representation of llms for text-to-image generation. In *AAAI*, 2025a.

Mushui Liu, Dong She, Jingxuan Pang, Qihan Huang, Jiacheng Ying, Wanggui He, Yuanlei Hou, and Siming Fu. Tfcustom: Customized image generation with time-aware frequency feature guidance. In *CVPR*, 2025b.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *International Conference on Learning Representations*, volume 2024, pp. 1862–1874, 2024.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022.

Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J Liu, James Harrison, Jaehoon Lee, Kelvin Xu, et al. Beyond human data: Scaling self-training for problem-solving with language models. *arXiv preprint arXiv:2312.06585*, 2023.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Yunze Tong, Mushui Liu, Canyu Zhao, Wanggui He, Shiyi Zhang, Hongwei Zhang, Peng Zhang, Jinlong Liu, Ju Huang, Jiamang Wang, et al. Alleviating sparse rewards by modeling step-wise and long-term sampling effects in flow-based grpo. *arXiv preprint arXiv:2602.06422*, 2026.

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 *CVPR*, pp. 8228–8238, 2024.

Jing Wang, Jiajun Liang, Jie Liu, Henglin Liu, Gongye Liu, Jun Zheng, Wanyuan Pang, Ao Ma, Zhenyu Xie, Xintao Wang, et al. Grpo-guard: Mitigating implicit over-optimization in flow matching via regulated clipping. In *CVPR*, pp. 5988–5998, 2026.

Fangtai Wu, Hailong Guo, Shijie Huang, Jiayi Song, Yubo Huang, Mushui Liu, Zhao Wang, Yunlong Yu, Jiaming Liu, and Ruihua Huang. Collectionlora: Collecting 50 effects in 1 lora via multi-teacher on-policy distillation. *arXiv preprint arXiv:2605.25378*, 2026.

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

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. In *NeurIPS*, pp. 15903–15935, 2023.

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.

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*, 2025b.

Yukang Yang, Dongnan Gui, Yuhui Yuan, Weicong Liang, Haisong Ding, Han Hu, and Kai Chen. Glyphcontrol: Glyph conditional control for visual text generation. *Advances in Neural Information Processing Systems*, 36:44050–44066, 2023.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T Freeman. Improved distribution matching distillation for fast image synthesis. In *NeurIPS*, pp. 47455–47487, 2024a.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *CVPR*, pp. 6613–6623. IEEE, 2024b.

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.

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 *ICLR*, pp. 134129–134150, 2026.

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

This supplementary material provides additional evaluation details, a theoretical proof, extended analysis, and qualitative results referenced in the main paper. The contents are organized as follows:

- Appendix A outlines the mathematical foundations of Flow Matching and SDE transitions, derives the affine relationship between next-state predictions and the velocity field, and gives the transition-to-velocity conversion underlying the Self-OPD loss.
- Appendix B proves Proposition 1, deriving the per-step KL gradient that motivates the transition-variance normalization in the Self-OPD loss.
- Appendix C derives the joint reward-tilted target and defines cross-dataset preference consistency.
- Appendix D documents the evaluation metrics, including the two GenEval scoring modes and the two preference evaluation protocols.
- Appendix E gives additional qualitative results demonstrating the aesthetic advantage of Self-OPD on the task-specific prompts.

A Preliminaries

A.1 Flow Matching and SDE Formulation

Flow Matching (FM) learns a continuous velocity field v_θ that transports a noise distribution $p_1 = \mathcal{N}(0, \mathbf{I})$ to the data distribution p_0 along straight-line trajectories. Formally, the probability path is defined as $x_t = (1-t)x_0 + t\epsilon$ for $t \in [0, 1]$, where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$. The model parameterizes $v_\theta(x_t, t)$ by minimizing the flow matching objective:

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}_{t,x_0,\epsilon}\left[\|v_\theta(x_t, t) - (\epsilon - x_0)\|^2\right]. \tag{10}$$

During inference, samples are generated by integrating the reverse-time ordinary differential equation (ODE) $\frac{\mathrm{d}x_t}{\mathrm{d}t} = v_\theta(x_t, t)$. Specifically, we discretize the trajectory into S integration steps along a discrete schedule $\{t_j\}_{j=0}^{S}$.

To enable stochastic exploration, the reverse ODE can be augmented into a reverse-time stochastic differential equation (SDE) Song et al. (2020); Fang et al. (2026).

$$x_{t_{j+1}} = x_{t_{j+1},\theta} + \sigma_{t_j}\sqrt{|\Delta t_j|}\, z_j, \quad z_j \sim \mathcal{N}(0, \mathbf{I}), \tag{11}$$

where $\sigma_{t_j} = \eta\sqrt{t_j/(1-t_j)}$ represents the noise schedule parameterized by the coefficient $\eta \in [0, 1]$. Conditioned on a data point x_0, the conditional distribution is $x_t \mid x_0 \sim \mathcal{N}\big((1-t)x_0,\, t^2\mathbf{I}\big)$, which yields the conditional score $\nabla_{x_t} \log p_t(x_t \mid x_0) = -\frac{x_t-(1-t)x_0}{t^2} = -\frac{\epsilon}{t}$. Averaging over the posterior $p(x_0 \mid x_t)$ establishes the marginal-score identity $\nabla_{x_t} \log p_t(x_t) = -\frac{\mathbb{E}[\epsilon|x_t]}{t}$. By eliminating $x_0 = \frac{x_t - t\epsilon}{1-t}$ from the parameterized velocity target $v_\theta \approx \mathbb{E}[\epsilon - x_0 \mid x_t]$, we obtain the conditional expectation $\mathbb{E}[\epsilon \mid x_t] = x_t + (1-t)v_\theta$, which simplifies the score function to:

$$\nabla_{x_t} \log p_t(x_t) \approx -\frac{x_t + (1-t)\, v_\theta(x_t, t)}{t}. \tag{12}$$

The reverse SDE transition follows the score-augmented drift $v_\theta - \frac{\sigma_{t_j}^2}{2}\nabla_{x_t} \log p_t$ Song et al. (2020). Substituting Eq. 12 into this drift formulation scales the correction term to $\frac{\sigma_{t_j}^2}{2t_j}\big(x_{t_j} + (1-t_j)v_\theta\big)$. Consequently, the deterministic next-state prediction $x_{t_{j+1},\theta}$ is formulated as:

$$x_{t_{j+1},\theta} = x_{t_j} + \left[v_\theta(x_{t_j}, t_j) + \frac{\sigma_{t_j}^2}{2t_j}\Big(x_{t_j} + (1-t_j)\, v_\theta(x_{t_j}, t_j)\Big)\right]\Delta t_j. \tag{13}$$

Setting $\eta = 0$ naturally recovers the deterministic Euler ODE step $x_{t_j} + v_\theta \Delta t_j$. To explicitly reveal the dependence of the prediction on the learned velocity, we collect the v_θ terms in Eq. 13, yielding the coefficient $\Delta t_j\big[1 + \frac{\sigma_{t_j}^2}{2t_j}(1-t_j)\big]$. Substituting the schedule $\sigma_{t_j}^2 = \eta^2 t_j/(1-t_j)$ cancels

the terms t_j and $(1-t_j)$, leaving a timestep-independent constant $\big(1+\frac{\eta^2}{2}\big)$. Hence, the deterministic prediction displays an elegant affine relationship with the velocity field:

$$x_{t_{j+1},\theta} = b_{t_j}(x_{t_j}) + c_{t_j}\, v_\theta(x_{t_j}, t_j), \qquad c_{t_j} = \big(1+\tfrac{\eta^2}{2}\big)\Delta t_j, \tag{14}$$

where $b_{t_j}(x_{t_j}) = \big(1+\frac{\sigma_{t_j}^2 \Delta t_j}{2t_j}\big)x_{t_j}$ aggregates all components independent of v_θ. This affine equivalence guarantees that any distillation objective formulated in transition space can be minimized with identical convergence properties in velocity space.

A.2 Transition-to-Velocity Conversion

Because $x_{t_{j+1},\theta}$ is affine in v_θ by Eq. 14, every branch can be associated with an effective branch velocity $v^{(k)}$ satisfying $x^{(k)}_{t_{j+1}} = b_{t_j}(x_{t_j}) + c_{t_j} v^{(k)}$. Subtracting Eq. 14 from the branch definition Eq. 1 and dividing by $c_{t_j} = (1+\eta^2/2)\Delta t_j$ gives the closed form

$$v^{(k)} = v_\theta - \frac{\sigma_{t_j}}{(1+\eta^2/2)\sqrt{|\Delta t_j|}}\, z_k, \tag{15}$$

so injecting Gaussian noise in transition space is exactly a perturbation of the velocity target. Consequently, the displacement factors as $x_{t_{j+1},\theta} - x^{(k)}_{t_{j+1}} = c_{t_j}\,(v_\theta - v^{(k)})$, and the squared transition distance equals the squared velocity distance scaled by $c_{t_j}^2 = (1+\eta^2/2)^2|\Delta t_j|^2$. Dividing by the transition-kernel variance normalization $2\sigma_{t_j}^2|\Delta t_j|$ yields the prefactor $(1+\eta^2/2)^2|\Delta t_j|/(2\sigma_{t_j}^2)$ used in Eq. 5.

B Per-Step KL Analysis

This section provides the full derivation of Proposition 1, the per-step KL gradient invoked in Sec. 3.2. It shows that differentiating the reverse KL between the reward-tilted target q^* and the student transition kernel q_θ with respect to the student transition mean yields the precision-weighted mean displacement of Eq. 6, which is precisely what justifies the transition-variance normalization $1/(\sigma_{t_j}^2|\Delta t_j|)$ in the Self-OPD loss (Eq. 5).

B.1 KL Gradient Derivation

Abbreviate the transition mean $\mu_\theta = x_{t_{j+1},\theta}$ and the isotropic covariance $\Sigma = \sigma_{t_j}^2|\Delta t_j|\, I$. Since q^* is the fixed regression target, differentiating the reverse KL with respect to μ_θ acts only through the $\log q_\theta$ term. Splitting the two terms,

$$D_{\mathrm{KL}}(q^*\|q_\theta) = \mathbb{E}_{x\sim q^*}[\log q^*(x)] - \mathbb{E}_{x\sim q^*}[\log q_\theta(x)], \tag{16}$$

the first is independent of μ_θ and vanishes under ∇_{μ_θ}. For the Gaussian kernel,

$$\log q_\theta(x) = -\tfrac{1}{2}(x-\mu_\theta)^\top \Sigma^{-1}(x-\mu_\theta) - \tfrac{1}{2}\log\big((2\pi)^d|\Sigma|\big), \tag{17}$$

only the quadratic form depends on μ_θ, so $\nabla_{\mu_\theta}\log q_\theta(x) = \Sigma^{-1}(x-\mu_\theta)$. By linearity of expectation,

$$\nabla_{\mu_\theta} D_{\mathrm{KL}}(q^*\|q_\theta) = -\,\mathbb{E}_{x\sim q^*}\big[\Sigma^{-1}(x-\mu_\theta)\big] = \Sigma^{-1}\,\mathbb{E}_{x\sim q^*}\big[\mu_\theta - x\big], \tag{18}$$

and substituting $\Sigma^{-1} = \frac{1}{\sigma_{t_j}^2|\Delta t_j|} I$ recovers Eq. 6. □

B.2 From the KL Gradient to the Training Loss

The gradient has a direct operational reading: the KL-optimal update moves the student mean $x_{t_{j+1},\theta}$ toward samples of the reward-tilted target q^* with a step size set by the transition *precision* $1/(\sigma_{t_j}^2|\Delta t_j|)$. Because q^* is intractable, Self-OPD uses the K sampled SDE branches to construct a practical signed pull–push surrogate, weighting branches by their advantages A_k and attenuating negative branches with the direction gate d_k. Rewriting the mean-space displacement as a velocity-space displacement through the affine relation in Eq. 14 yields Eq. 5, including the constant $(1+\eta^2/2)^2$ derived in Appendix A.2. Thus, the KL identity motivates the transition-precision normalization, while the signed advantage weighting and direction-aware attenuation define our practical multi-branch surrogate.

C Reward-Level Fusion Derivation

C.1 Joint-Target Normalization

Let $q_{\bar{\theta}}(\cdot \mid x_{t_j})$ denote the fixed student transition kernel used to sample the current branches, where $\bar{\theta}$ is detached during target construction. Let A_m denote objective m's contribution to the branch advantage after per-reward standardization and the shared normalization in Eq. 3, such that $A = \sum_m \lambda_m A_m$. The reward-tilted target for objective m is

$$q_m^*(x_{t_{j+1}} \mid x_{t_j}) = \frac{1}{Z_m} q_{\bar{\theta}}(x_{t_{j+1}} \mid x_{t_j}) \exp\left(\frac{A_m(x_{t_{j+1}})}{\tau}\right), \tag{19}$$

where Z_m is independent of $x_{t_{j+1}}$. Aggregating the density ratios relative to the same sampling kernel gives

$$\begin{aligned} q_{\text{joint}}^*(x_{t_{j+1}} \mid x_{t_j}) &\propto q_{\bar{\theta}}(x_{t_{j+1}} \mid x_{t_j}) \prod_{m=1}^{M} \left(\frac{q_m^*(x_{t_{j+1}} \mid x_{t_j})}{q_{\bar{\theta}}(x_{t_{j+1}} \mid x_{t_j})}\right)^{\lambda_m} \\ &\propto q_{\bar{\theta}}(x_{t_{j+1}} \mid x_{t_j}) \exp\left(\frac{1}{\tau} \sum_{m=1}^{M} \lambda_m A_m(x_{t_{j+1}})\right). \end{aligned} \tag{20}$$

The product of normalizers $\prod_m Z_m^{-\lambda_m}$ is independent of $x_{t_{j+1}}$ and is absorbed into proportionality. The M rewards therefore define one tilt through the composite advantage $\sum_m \lambda_m A_m$. This is a weighted geometric aggregation of density ratios; it does not require $\sum_m \lambda_m = 1$. It favors transitions with a high weighted aggregate reward, but does not by itself guarantee that every component reward is individually maximized.

The implementation follows this construction through Eq. 9. Let $s > 0$ denote the shared denominator in Eq. 3 and define $A_m^{(k)} = (\tilde{r}_m^{(k)} - \tilde{r}_m^{\text{ode}})/s$. Then $A_k = \sum_m \lambda_m A_m^{(k)}$. Consequently, the composite reward induces the same branch ordering as the composite advantage. Self-OPD uses this score to rank and weight all sampled branches; it does not select only one branch or differentiate through the reward models.

C.2 Trajectory Selection versus Teacher Velocity/Transition Matching

The composite score $r^{(k)}$ enters training only through the branch ranking that defines A_k; it is never differentiated. Consequently, the target in Eq. 5 remains a single, concrete trajectory velocity $v_{\pm}^{(k)}$ evaluated jointly by the applicable metrics. Flow-OPD transfers one reward-specialized teacher through velocity or transition matching, while DiffusionOPD extends this mechanism to task-conditioned integration of multiple teachers. In both cases, teacher supervision is specified before the output is evaluated. Reward-level fusion instead combines scores after sampling and ranks complete trajectories according to evaluations of the same image. This construction accommodates black-box scorers and allows the reward trade-off weights $\{\lambda_m\}$ to be adjusted without differentiating through the reward models.

C.3 Cross-Dataset Preference Consistency

For a preference metric s, we measure how much preference quality changes between aesthetic prompts and task prompts by

$$\Delta_s = \left|\mathbb{E}_{c \sim \mathcal{D}_{\text{aes}}} s(G_\theta(c)) - \mathbb{E}_{c \sim \mathcal{D}_{\text{task}}} s(G_\theta(c))\right|, \tag{21}$$

where G_θ is the generator, $\mathcal{D}_{\text{aes}}$ is the aesthetic prompt set, and $\mathcal{D}_{\text{task}}$ contains the GenEval/OCR prompts. A smaller Δ_s indicates that preference quality varies less across prompt distributions. As shown in Fig. 4(e,f), Self-OPD has smaller PickScore and HPSv2 gaps than DiffusionOPD and its aesthetic co-teacher control. This observation supports, but does not by itself prove, the interpretation that reward-level fusion better preserves preference quality across prompt families.



Figure 5: **Strict vs. continuous scoring** on "a photo of a white handbag and a red giraffe". (a) base; (b) Self-OPD.

Model	Strict	Continuous
SD3.5-Medium (base)	0	0.5
Self-OPD	1	1.0

Table 3: Strict and continuous GenEval scores for the two images in Fig. 5. The prompt has two sub-requirements (one per colored object).

D Evaluation Metrics

D.1 GenEval: Strict vs. Continuous Scoring

GenEval Ghosh et al. (2023) evaluates compositional generation by checking whether generated images satisfy a set of sub-requirements (e.g., correct object count, spatial relation, color attribute). We report two scoring modes:

- **Strict scoring.** A binary per-image metric: the score is 1 only if *all* sub-requirements for that prompt are satisfied, and 0 otherwise. The reported number is the average over all test prompts. This is a demanding metric that penalizes any partial failure.
- **Continuous scoring.** A partial-credit metric: for each image, the score is the fraction of sub-requirements satisfied (ranging from 0 to 1). The reported number is again averaged over all test prompts. This gives credit to images that partially fulfill the prompt.

Fig. 5 illustrates both modes on the color-attribution prompt "a photo of a white handbag and a red giraffe", which has two sub-requirements, one per colored object. The SD3.5-Medium base (Fig. 5a) renders the white handbag correctly but colors the giraffe orange rather than red, satisfying one of the two sub-requirements: its strict score is 0 while its continuous score is 0.5. Self-OPD (Fig. 5b) renders both the white handbag and the red giraffe, satisfying both sub-requirements for a strict score of 1 and a continuous score of 1.0 (Tab. 3). Continuous scoring better captures incremental improvements, while strict scoring reflects end-to-end correctness.

D.2 Preference Protocols: Same Test Images vs. Separate Test Set

We evaluate preference-based metrics (PickScore Kirstain et al. (2023) and HPSv2 Wu et al. (2023)) under two complementary protocols:

- **Same test images.** PickScore and HPSv2 are computed on the *same* images generated for the GenEval and OCR test prompts. Specifically, for each model we generate images for the GenEval test suite and the OCR test suite, then score all these images with PickScore and HPSv2. The final score is the unweighted average of the GenEval-prompt score and the OCR-prompt score (1:1 averaging). This protocol measures whether a model maintains aesthetic quality and human preference *on the task it was trained for*, without introducing any distribution shift from additional prompts.

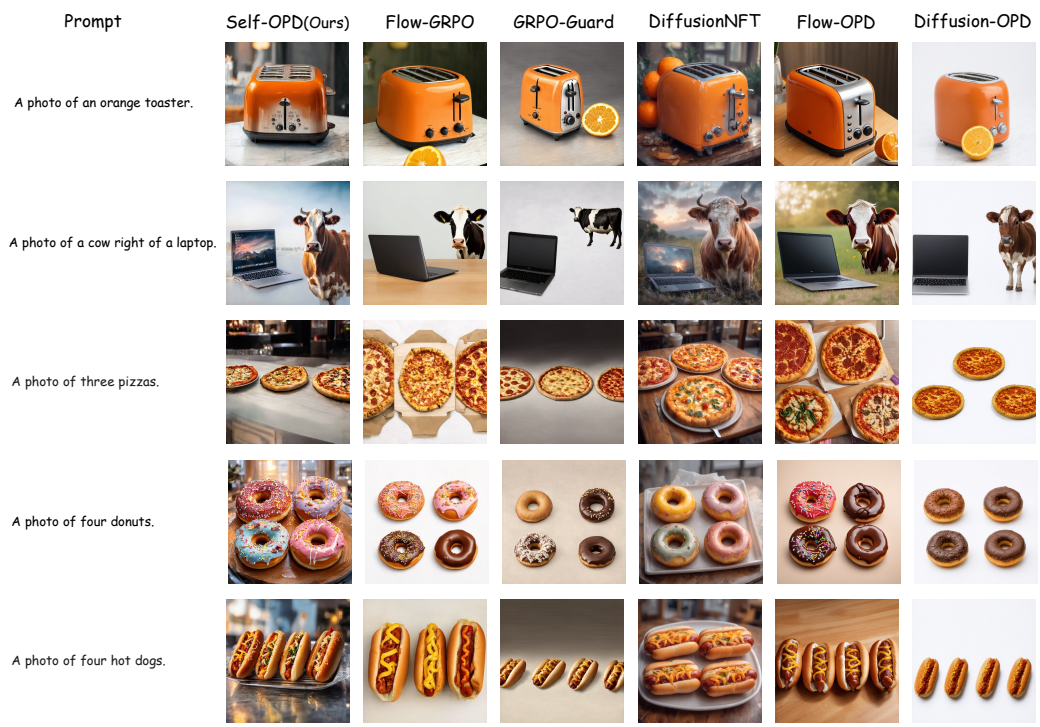

Figure 6: **Aesthetic quality on GenEval prompts.** Each method uses its multi-reward model if available, or its GenEval-trained model otherwise (Flow-GRPO, GRPO-Guard). Self-OPD satisfies the compositional requirements (counting, spatial relations, attributes) while generating images with higher aesthetic quality, consistent with its higher PickScore/HPSv2 in Tab. 2.

- **Separate test set.** PickScore and HPSv2 are computed on a held-out set of aesthetic-oriented prompts (DrawBench) that are disjoint from the training and GenEval/OCR evaluation prompts. This measures preference quality on a general-purpose prompt distribution.

The "same test images" protocol gives a more controlled cross-metric comparison because it evaluates all metrics on the same set of generations, separating model quality from prompt distribution. A model with high preference scores under this protocol improves aesthetic quality without sacrificing task performance. The "separate test set" protocol can instead favor models that overfit to aesthetic-leaning prompts while degrading on structured tasks. For image generation we care about the overall quality of every generated image rather than separate high scores on different task sets, so we treat the "same test images" protocol as the primary preference comparison. Fig. 4(e,f) makes this concrete: DiffusionOPD's lead on the separate set does not carry over to the GenEval and OCR test images, because its task-conditioned teacher-field integration leaves preference quality coupled to the prompt family, whereas Self-OPD stays high on both.

E Additional Qualitative Results

Fig. 6 and Fig. 7 visualize the aesthetic quality gap on task-specific prompts, comparing Self-OPD against Flow-GRPO, GRPO-Guard, DiffusionNFT, Flow-OPD, and DiffusionOPD. Under the same test images protocol, Self-OPD attains the highest PickScore and HPSv2 among all methods (Tab. 2); these figures illustrate why. On GenEval prompts (Fig. 6), Self-OPD produces images with richer lighting, more natural textures, and better composition while satisfying the compositional requirements.

As shown in Fig. 7, Self-OPD renders text accurately and simultaneously generates more visually appealing scenes with coherent backgrounds and realistic details, whereas other methods tend to produce flatter or overly simplistic images. Given identical prompts, Self-OPD maintains high aesthetic

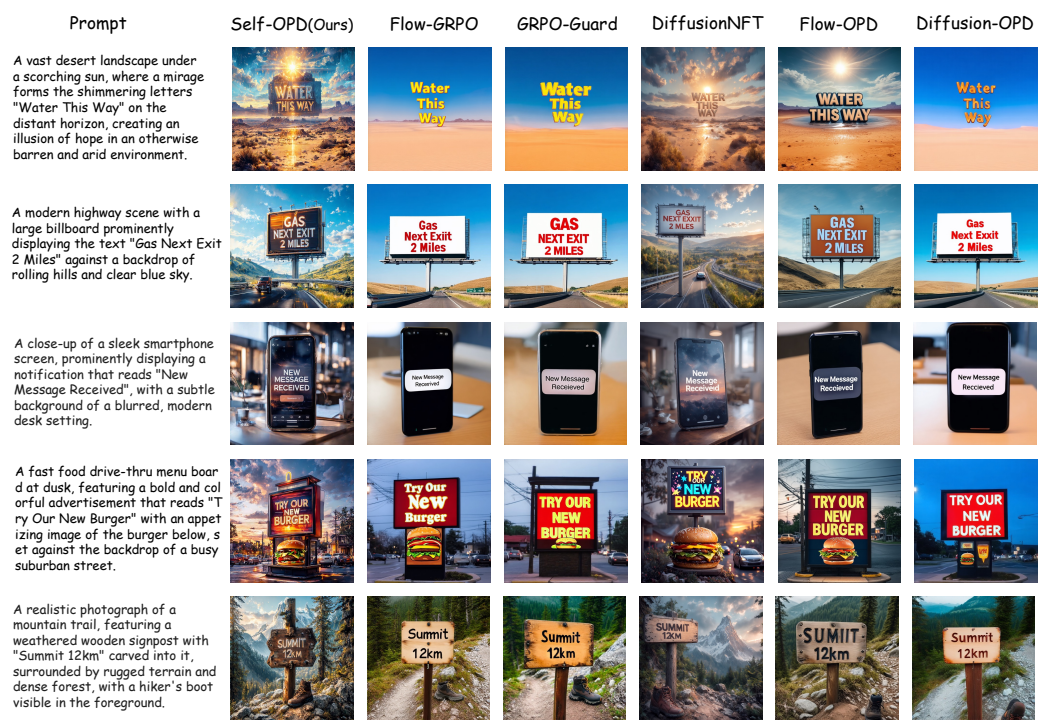

Figure 7: **Aesthetic quality on OCR prompts.** Each method uses its multi-reward model if available, or its OCR-trained model otherwise (Flow-GRPO, GRPO-Guard). Self-OPD renders the target text accurately while producing more visually appealing scenes with richer backgrounds and realistic details.

quality, while DiffusionOPD and the RL baselines exhibit visible degradation, e.g., over-saturation and loss of detail.