DIFFUSION REINFORCEMENT LEARNING VIA CENTERED REWARD DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion and flow models achieve State-Of-The-Art (SOTA) generative performance, yet many practically important behaviors such as fine-grained prompt fidelity, compositional correctness, and text rendering are weakly specified by score or flow matching pretraining objectives. Reinforcement Learning (RL) fine-tuning with external, black-box rewards is a natural remedy, but diffusion RL is often brittle. Trajectory-based methods incur high memory cost and high-variance gradient estimates; forward-process approaches converge faster but can suffer from distribution drift, and hence reward hacking. In this work, we present **Centered Reward Distillation (CRD)**, a diffusion RL framework derived from KL-regularized reward maximization built on forward-process-based fine-tuning. The key insight is that the intractable normalizing constant cancels under *within-prompt centering*, yielding a well-posed reward-matching objective. To enable reliable text-to-image fine-tuning, we introduce techniques that explicitly control distribution drift: (*i*) decoupling the sampler from the moving reference to prevent ratio-signal collapse, (*ii*) KL anchoring to a CFG-guided pretrained model to control long-run drift and align with the inference-time semantics of the pre-trained model, and (*iii*) reward-adaptive KL strength to accelerate early learning under large KL regularization while reducing late-stage exploitation of reward-model loopholes. Experiments on text-to-image post-training with `GenEval` and `OCR` rewards show that CRD achieves competitive reward optimization results with fast convergence and reduced reward hacking, as validated on unseen preference metrics.

1 INTRODUCTION

Diffusion and flow models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song & Ermon, 2019; Song et al., 2020; Vahdat et al., 2021; Karras et al., 2022; Lipman et al., 2022; Albergo et al., 2023; Liu et al., 2022; Liu, 2025) have become a cornerstone of modern generative modeling, achieving state-of-the-art performance across modalities and tasks (Ho et al., 2022; Rombach et al., 2022; Peebles & Xie, 2023; Dufour et al., 2024; Wang et al., 2025b; Degeorge et al., 2025; Boudier et al., 2025; Courant et al., 2025; Esser et al., 2024; Wan et al., 2025). Yet many behaviors that matter most in practice, such as aesthetic quality, fine-grained prompt fidelity, and legible text rendering, are only weakly specified by denoising score matching (Vincent, 2011; Song & Ermon, 2019) objectives on static datasets, and are therefore not reliably induced during pretraining. Closing this gap calls for *post-training* strategies that optimize models with respect to external signals. Reinforcement Learning (RL) is particularly well-suited to this end: rather than matching a fixed data distribution, the model is explicitly optimized toward downstream objectives using rewards derived from human preference (Wu et al., 2023; Ma et al., 2025b; Xu et al., 2023; Schuhmann, 2022), vision-language evaluators (Hessel et al., 2021), or task-specific metrics (Chen et al., 2023; Ghosh et al., 2023).

Diffusion RL aims to *post-train* a pretrained model to maximize external reward signals while preventing excessive drift from the original distribution. Early approaches either operated off-policy with expensive data collection, or required differentiable rewards (*e.g.*, `CLIPScore` (Hessel et al., 2021)) to backpropagate through the denoising process (Peng et al., 2019; Clark et al., 2023; Fan & Lee, 2023; Black et al., 2023; Lee et al., 2023; Yuan et al., 2024), excluding rewards that are non-differentiable or expensive to differentiate through (*e.g.*, `OCR` accuracy, VLM-based evaluators). Recent diffusion RL work has pursued two main directions.

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.

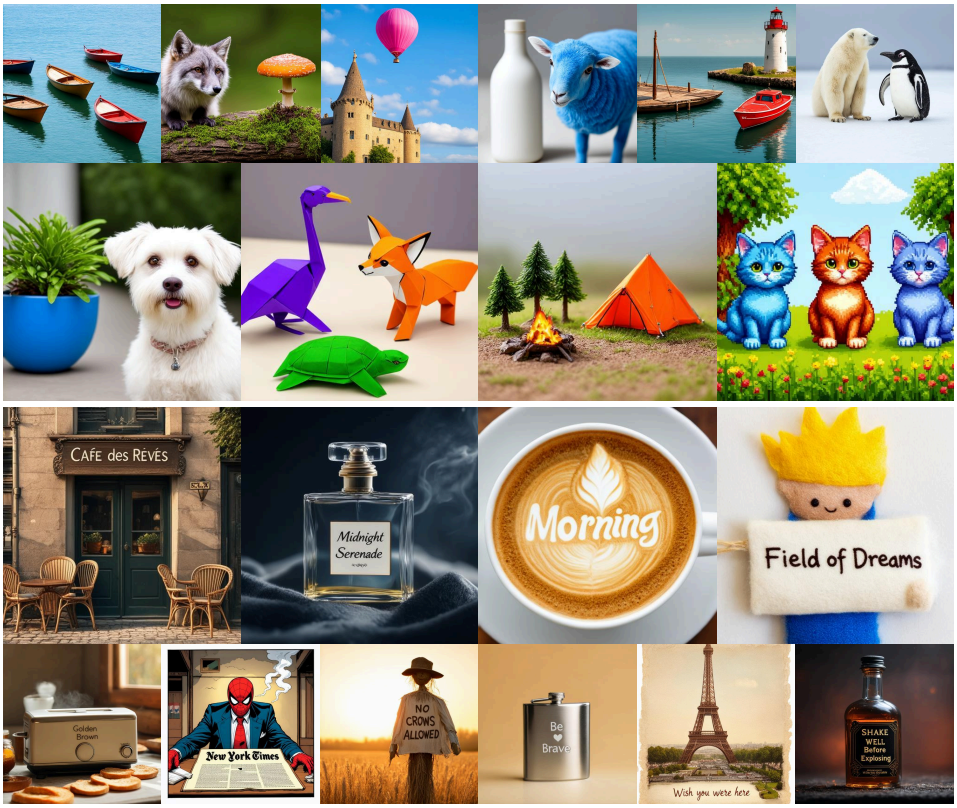


Figure 1: Qualitative results produced by our RL fine-tuned SD3.5M (Esser et al., 2024) model with `GenEval` (Ghosh et al., 2023) reward (top) and `OCR` (Chen et al., 2023) reward (bottom).

First, GRPO-style methods (Shao et al., 2024; Guo et al., 2025) (e.g., Flow-GRPO (Liu et al., 2025b; Xue et al., 2025b) and follow-ups (Li et al., 2025; Ding & Ye, 2025; Deng et al., 2026; Liu et al., 2025a; Chen et al., 2025c)) view denoising as an explicit Markov Decision Process (MDP) whose rollout is the full SDE sampling trajectory, enabling principled credit assignment and importing the RLHF toolkit developed for autoregressive LLMs (Guo et al., 2025; Yu et al., 2025; Liu et al., 2025c; Yue et al., 2025). However, storing full SDE trajectories and high-variance training signals impose high memory and compute overhead that slows convergence, which is particularly prohibitive for large-scale diffusion models (Wan et al., 2025).

A second line of work (Xue et al., 2025a; Zheng et al., 2025; Luo et al., 2025b; Wang et al., 2026b; Choi et al., 2026; Chen et al., 2025d; Ou et al., 2026) formulates diffusion RL via the *forward process* by *decoupling sampling from training states* (Zheng et al., 2025): clean samples are generated from the current model and training proceeds on their forward-noised versions, yielding objectives that resemble *advantage-weighted maximum likelihood*. These objectives are simpler to implement and substantially more efficient, with lower-variance gradients that often converge faster (Xue et al., 2025a). However, as a fixed reference model becomes a poor surrogate during training, these methods maintain a *moving reference* periodically updated toward the current model; if it tracks too aggressively, it enables iterative drift (Xue et al., 2025a; Zheng et al., 2025), which can amplify reward hacking by encouraging exploitation of reward imperfections.

In this work, we revisit the forward-process paradigm through the lens of KL-regularized reward maximization, where the optimal solution is a reward-tilted (Peters & Schaal, 2007; Peng et al., 2019; Korbak et al., 2022; Go et al., 2023; Rafailov et al., 2023; Uehara et al., 2025; Kim et al., 2025; Pachebat et al., 2025; Sabour et al., 2025; Potaptchik et al., 2025) version of a reference model. This perspective implies that external rewards correspond to a scaled log-density ratio between the optimal fine-tuned model and the reference model, but only *up to an unknown, prompt-dependent normalizer*. Because this normalizer is intractable, directly regressing model likelihood ratios toward absolute rewards is ill-posed (Mao et al., 2024; Gao et al., 2024; Fisch et al., 2024; Zhu et al., 2025a; Matrenok et al., 2025). Our key insight is that the normalizer cancels exactly under *within-prompt centering*: for a group of samples drawn under the same prompt, the unknown normalizer term is identical across samples and disappears when rewards are centered within the group. Building on this observation, we propose **Centered Reward Distillation (CRD)**, a set of well-posed reward-

matching objectives for diffusion RL. CRD trains the diffusion model so that its implicit log-density ratio (approximated via a diffusion Evidence Lower BOund surrogate, ELBO) matches *centered* external rewards within each prompt group. This yields a unifying view that recovers prior works in LLM such as reward-distillation methods (Mao et al., 2024; Gao et al., 2024; Fisch et al., 2024) and GVPO objective (Zhang et al., 2025) as special cases, and naturally extends to a ratio-based variant connected to InfoNCA (Chen et al., 2024a).

To make CRD reliable for moving reference RL fine-tuning, we introduce practical stabilization techniques that explicitly mitigate reward hacking by controlling distribution drift. First, we decouple the model used to generate samples from the *moving reference* used in the log-ratio objective, which prevents training instabilities when the reference drifts too close to the current model. Second, to control long-run drift, we add a KL penalty that anchors the current model to a *fixed* initial reference (the pretrained model). Importantly, because many forward-process setups train and sample the current model *without* Classifier-Free Guidance (CFG) (Ho & Salimans, 2022; Wang et al., 2024; Kynkäänniemi et al., 2024) for efficiency, we adopt the *CFG-guided* version of the initial reference model, aligning regularization with inference-time semantics (and reducing to CFG distillation when the RL signal is absent). Finally, we use a simple reward-adaptive scaling of the anchoring strength to accelerate early learning while preventing late-stage exploitation of reward-model loopholes.

We evaluate CRD on text-to-image RL fine-tuning with `GenEval` and `OCR`-based rewards, comparing against representative Flow-GRPO-style and forward-process baselines. Overall, our approach achieves competitive reward optimization with faster and more stable training, and reduced reward hacking behavior.

Our contributions can be summarized as follows:

- We introduce **CRD**, a within-prompt centered reward-matching framework that removes the unknown prompt-dependent normalizer and recovers prior reward distillation and GVPO-style objectives as special cases.
- We propose practical techniques for stable RL fine-tuning: decoupled sampling, KL anchoring to a *CFG*-guided fixed reference, and reward-adaptive KL strength.
- Experiments on `GenEval` and `OCR` rewards show competitive performance against recent diffusion RL baselines, with improved stability and reduced reward hacking.

2 Related Works

RL for diffusion models. Progress in diffusion and flow-based generative modeling has motivated works that apply RL as a *post-training* stage to better align generations with human preferences or task-specific reward signals (Peng et al., 2019; Clark et al., 2023; Fan & Lee, 2023; Black et al., 2023; Lee et al., 2023; Yuan et al., 2024). Recently, *Group Relative Policy Optimization* (GRPO) (Shao et al., 2024; Guo et al., 2025) has shown strong alignment performance in diffusion and flow-based models (Liu et al., 2025b; Xue et al., 2025b), and follow-up works improve it through trajectory structuring for exploration and credit assignment (He et al., 2025c; Li et al., 2025; Ding & Ye, 2025; Luo et al., 2025a), dense rewards and ratio/regularization stabilizers (Deng et al., 2026; He et al., 2025b; Wang et al., 2025a), more efficient and faithful rollouts (Chen et al., 2025c; Wang & Yu, 2025; Zhang et al., 2026; He et al., 2025a), and richer reward definitions (Mao et al., 2025; Liu et al., 2025a; Wang et al., 2026a; Zhou et al., 2025). However, most of these methods optimize through the *backward-process* MDP formulation, which introduces timestep-dependent variance, rollout overhead, and credit-assignment challenges that ultimately limit training efficiency.

Diffusion RL based on the forward process. A recent line of work instead formulates diffusion RL directly through the *forward process* (Xue et al., 2025a; Zheng et al., 2025; Luo et al., 2025b; Wang et al., 2026b; Choi et al., 2026; Chen et al., 2025d; Zhu et al., 2025b): clean samples are generated without storing intermediate noisy states, and the loss is computed on their forward-noised versions, which these works report to train significantly faster than backward-process methods such as Flow-GRPO. GPO (Chen et al., 2025d) interprets an ELBO-based objective as an *advantage-weighted log-likelihood*; AWM (Xue et al., 2025a) proposes *Advantage Weighted Matching* to mitigate the variance of Flow-GRPO-style estimators; DiffusionNFT (Zheng et al., 2025) derives an update toward an optimal velocity direction while using fast ODE samplers; DGPO (Luo et al.,

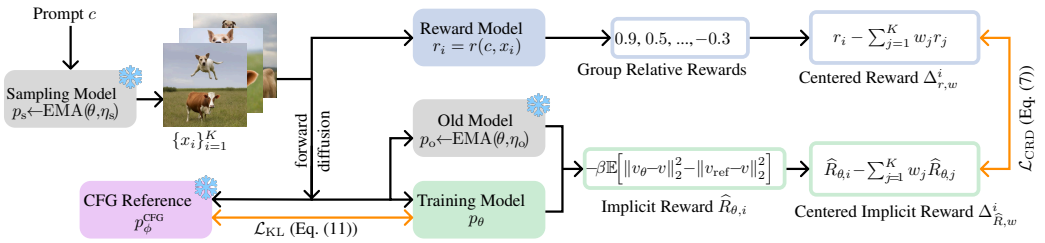


Figure 2: For each prompt, K samples $\{x_i\}_{i=1}^K$ are generated from the sampling model p_{samp} (p_s). A reward model produces external rewards $r(c, x_i)$, and implicit model rewards $\widehat{R}_{\theta,i}$ are estimated via diffusion ELBO differences between the current model p_θ and a moving reference p_{old} (p_o). Within-prompt centering yields $\Delta^i_{r,w}$ and $\Delta^i_{\widehat{R},w}$, cancelling the prompt-dependent normalizer and enabling a well-posed matching objective. An initial KL penalty with respect to the fixed CFG-guided pretrained model p_ϕ^{CFG} is imposed to prevent reward hacking.

2025b) learns from group-level preferences; GDRO (Wang et al., 2026b) derives a cross-entropy objective closely related to InfoNCA (Chen et al., 2024a); and Choi et al. (2026) systematically study design choices in this family. Our method falls within this forward-process RL family and inherits its computational efficiency, while introducing a new class of objectives and practical techniques that prevent reward hacking without sacrificing training efficiency. An extended discussion of related works is provided in Appendix B.

3 Diffusion RL using Reward Distillation

We first introduce the KL-regularized formulation of diffusion RL and show that rewards correspond to a log-density ratio only up to an unknown prompt-dependent normalizer (Sec. 3.1). We then present **Centered Reward Distillation (CRD)**, which removes this normalizer via within-prompt centering and yields a family of reward-matching objectives (Sec. 3.2). Finally, we describe practical techniques that mitigate reward hacking under distribution drift (Sec. 3.3). The overall training pipeline is illustrated in Fig. 2 and the full algorithm is summarized in Algorithm 1 (Appendix C).

3.1 Background

Setup and goal. Given access to a fixed external reward $r(c, x)$ (e.g., from a reward model) and a pretrained diffusion model with parameters ϕ, the goal of diffusion RL is to fine-tune a copy of this model with parameters θ to maximize the expected reward $r(c, x)$ of its generated samples, while keeping the distribution $p_\theta(x|c)$ close to a reference distribution $p_{\text{ref}}(x|c)$ (which is usually the pretrained model $p_\phi(x|c)$) for each prompt condition c.

KL-regularized form and the unknown normalizer. Under the standard KL-regularized reward maximization objective, the optimal distribution satisfies (Peters & Schaal, 2007; Peng et al., 2019; Korbak et al., 2022; Go et al., 2023; Rafailov et al., 2023)

$$p_{\theta^*}(x \mid c) \;\propto\; p_{\text{ref}}(x \mid c) \exp\left(\frac{r(c, x)}{\beta}\right), \tag{1}$$

which implies

$$r(c, x) \;=\; \beta \log \frac{p_{\theta^*}(x \mid c)}{p_{\text{ref}}(x \mid c)} + \beta \log Z(c), \tag{2}$$

where $\beta > 0$ is the KL regularization strength and $Z(c) = \int p_{\text{ref}}(x|c) \exp(\frac{r(c,x)}{\beta}) dx$ is a prompt-dependent normalizer.

Implicit model reward and diffusion-specific surrogate. The *implicit model reward* is defined as the scaled log-density ratio:

$$R_\theta(c, x) \;\triangleq\; \beta \log \frac{p_\theta(x \mid c)}{p_{\text{ref}}(x \mid c)}. \tag{3}$$

For diffusion models, the exact log-density ratio in Eq. (3) is intractable. Following Wallace et al. (2024); McAllister et al. (2025); Ren et al. (2024); Choi et al. (2026), we use the diffusion ELBO surrogate to estimate this term:

$$\widehat{R}_\theta(c, x) \triangleq -\beta\, \mathbb{E}_{t,\epsilon}\left[w(t)(\|v_\theta(x_t, t|c) - v_{\text{target}}\|_2^2 - \|v_{\text{ref}}(x_t, t|c) - v_{\text{target}}\|_2^2)\right], \tag{4}$$

where t is sampled from a predefined timestep distribution, $\epsilon \sim \mathcal{N}(0, I)$, $w(t)$ is a time-dependent weighting which we set as 1 for simplicity, x_t is the noisy latent produced by the forward diffusion process from clean data x at time t, $v_{\text{target}}(\cdot) = \epsilon - x$ denotes the corresponding target velocity in flow matching objective and v_θ is the velocity predicted by the model θ. We use $\widehat{R}_\theta$ as an estimator of R_θ throughout what follows. Following previous work (Xue et al., 2025a; Zheng et al., 2025), we first transform the raw reward $r_{\text{raw}}(c, x)$ using group normalization to get $r(c, x)$.

Crucially, since the normalizer $\beta \log Z(c)$ is intractable, directly regressing $\widehat{R}_\theta(c, x)$ onto absolute rewards $r(c, x)$ is ill-posed without knowing the normalizer based on Eq. (2) (Zhu et al., 2025a; Matrenok et al., 2025).

3.2 Centered Reward Distillation (CRD)

Key insight: within-prompt centering (subtracting the weighted group reward mean) removes the normalizer $\beta \log Z(c)$. For each prompt c, we consider a group of K samples $\{x_i\}_{i=1}^K$ drawn from a proposal distribution and their corresponding external rewards $r(c, x_i)$. Recall from Eq. (2) that, under KL-regularized reward maximization, the reward decomposes as $r(c, x) = R_{\theta^*}(c, x) + \beta \log Z(c)$, where the normalizer term $\beta \log Z(c)$ depends only on the prompt c. Consequently, for a fixed prompt c and any weights $\{w_j\}_{j=1}^K$ satisfying $\sum_{j=1}^K w_j = 1$, the normalizer cancels under within-prompt centering:

$$r(c, x_i) - \sum_{j=1}^K w_j r(c, x_j) \;=\; R_{\theta^*}(c, x_i) - \sum_{j=1}^K w_j R_{\theta^*}(c, x_j). \tag{5}$$

This **centered matching** problem is well-posed regardless of the unknown normalizer, and directly motivates our objective: train the model so that its implicit reward $\widehat{R}_\theta(c, x)$ is consistent with the external rewards.

Reward-weighted centering weights. Within each prompt group, we define reward-weighted softmax weights: $w_i(c, \{x_j\}_{j=1}^K; \tau) \;\triangleq\; \frac{\exp(r(c,x_i)/\tau)}{\sum_{j=1}^K \exp(r(c,x_j)/\tau)}$ where $\tau > 0$ is temperature that controls the sharpness of the probability distribution over the estimated rewards. For brevity, we write $w_i \equiv w_i(c, \{x_j\}_{j=1}^K; \tau)$ throughout. We note that, as $\tau \to \infty$, $w_i \to 1/K$ (uniform centering), whereas as $\tau \to 0$, the weights concentrate on the highest-reward sample (max-anchored centering).

CRD objective. Let $\rho(c, \{x_i\}_{i=1}^K)$ be a distribution over prompts and the corresponding within-prompt generation sets with group size K (sampled from a proposal such as the current model (Rafailov et al., 2023; Ethayarajh et al., 2024; Azar et al., 2024), a mixture with p_{ref} (Gorbatovski et al., 2024), or a replay buffer (Xue et al., 2025a)). Motivated by the centering identity in Eq. (5), we define within-prompt centered residuals for both the reward model and the log-density ratio estimator:

$$\Delta^i_{r,w} \;\triangleq\; r(c, x_i) - \sum_{j=1}^K w_j\, r(c, x_j), \qquad \Delta^i_{\widehat{R},w} \;\triangleq\; \widehat{R}_\theta(c, x_i) - \sum_{j=1}^K w_j\, \widehat{R}_\theta(c, x_j). \tag{6}$$

The **CRD loss** then matches centered external rewards to centered implicit model reward by minimizing

$$\mathcal{L}^{(\tau)}_{\text{CRD}}(p_\theta; \rho) = \mathbb{E}_{\rho(c,\{x_i\})}\; \frac{1}{K} \sum_{i=1}^K \left(\Delta^i_{r,w} - \Delta^i_{\widehat{R},w}\right)^2. \tag{7}$$

Connections to existing methods. CRD can serve as a unifying framework that reinterprets several existing methods as special cases and admits natural extensions. We show in Appendix F that two-sample reward distillation (Mao et al., 2024; Gao et al., 2024; Fisch et al., 2024) methods and GVPO (Zhang et al., 2025) can both be recovered as special cases of CRD. In addition, in Appendix G we introduce a ratio-based distillation variant of CRD, and show that the InfoNCA loss proposed in Chen et al. (2024a) can be recovered as a special case under appropriate parameterization.

Training efficiency and stability: moving reference and decoupled sampling. As training progresses, a fixed pretrained reference can become misaligned with the current model, yielding noisy

log-density ratios and increasingly off-distribution samples. Following DiffusionNFT (Zheng et al., 2025), we maintain a *moving reference* p_{old}, held fixed within each epoch and updated at epoch end via Exponential Moving Average (EMA), so the reference and reference-sampled data remain close to p_θ. However, if p_{old} is updated too aggressively, it can drift too close to the current model p_θ, collapsing the log-ratio signal in Eq. (3) toward zero and destabilizing training. We therefore further decouple sampling from the log-ratio reference (see Fig. 2): samples are drawn from a separate EMA model p_{samp}, while p_{old} is used only in the log-density ratio and updated with a slower EMA. This keeps data collection near on-policy while preserving a meaningful log-ratio signal.

3.3 Reward-Adaptive CFG based KL Regularization

Motivation: reward hacking under online distribution drift. During online fine-tuning, the old model p_{old} evolves across iterations as p_θ is updated, inducing distribution drift away from the pretrained model. This distribution drift increases the risk of *reward hacking* (see Appendix E for more discussion): the model exploits weaknesses of the reward model by moving into out-of-distribution regions, leading to degraded sample quality despite high reward scores.

Mitigation: KL regularization to a fixed initial reference. A standard stabilization is to add a KL penalty that anchors the model θ to a *fixed* initial reference model (the pretrained model ϕ), preventing uncontrolled drift (Schulman et al., 2017; Ziegler et al., 2019; Ouyang et al., 2022):

$$\beta_{\text{init}}\text{KL}(p_\theta(\cdot \mid c) \parallel p_\phi(\cdot \mid c)) \approx \beta_{\text{init}}\mathbb{E}_{t,\epsilon}\Big[\lambda(t) \, \|v_\theta(x_t, t \mid c) - v_\phi(x_t, t \mid c)\|_2^2\Big], \tag{8}$$

where β_{init} denotes the strength of the initial KL regularization, $\lambda(t)$ is a time-dependent weight which we set to 1 for simplicity, and v_ϕ is the initial model velocity prediction. However, in the standard setup of forward-process based methods, the current model θ is sampled *without* CFG, while the conditional pretrained model (without CFG) can be significantly weaker than its CFG-guided counterpart. As a result, anchoring too strongly to a weak non-CFG reference can introduce a training mismatch: the KL term may dominate and pull the model toward low-quality regions of the pretrained conditional model, meanwhile counteracting reward optimization.

To align the anchoring distribution with the effective sampling semantics of the pretrained model, we instead use a KL penalty with the CFG-guided version of the pretrained model as the fixed reference:

$$\beta_{\text{init}}\text{KL}\big(p_\theta(\cdot \mid c) \parallel p_\phi^{\text{CFG}}(\cdot \mid c)\big), \tag{9}$$

where $p_\phi^{\text{CFG}}(\cdot \mid c)$ denotes the distribution induced by sampling the pretrained model ϕ with CFG (guidance scale s). This KL term is approximated analogously to Eq. (8), replacing v_ϕ with the CFG-guided velocity v_ϕ^{CFG}. For notation consistency, we rename the CRD coefficient β as β_{old} for moving reference.

Intuitively, this anchors training to a stronger, more reliable pretrained behavior, which reduces reward hacking while avoiding the degenerate pull toward the weak non-CFG conditional baseline. Note that when the RL gradient is missing, this initial KL regularization alone is functionally equivalent to CFG distillation (Meng et al., 2023; Chen et al., 2024b; 2025a; Cideron et al., 2024).

Reward-adaptive initial KL strength for faster optimization. While a large initial-reference KL coefficient improves stability, it can also slow learning by excessively restricting policy updates. Since our raw reward $r_{\text{raw}}(c, x)$ is a scalar and can be normalized to $[0, 1]$, we use a simple reward-adaptive scaling of the initial-reference KL strength:

$$\hat{\beta}_{\text{init}}(c, x) \;=\; r_{\text{raw}}(c, x)\, \beta_{\text{init}}, \qquad r_{\text{raw}}(c, x) \in [0, 1]. \tag{10}$$

We apply $\hat{\beta}_{\text{init}}(c, x)$ to the KL term in (9). This choice preserves nonnegativity, and it makes the anchoring weaker on low-reward samples (allowing larger corrective updates) and stronger on high-reward samples (preventing late-stage drift into reward-model loopholes), empirically accelerating training without sacrificing final performance. This strategy coincides with the takeaway of GARDO (He et al., 2025b). Since the adaptive KL strength is sample-dependent, we can finally write our KL loss as:

$$\mathcal{L}_{\text{KL}} = \frac{1}{K}\sum_{i=1}^{K} \hat{\beta}_{\text{init}}(c, x^i)\mathbb{E}_{t,\epsilon}\Big[\lambda(t) \, \big\|v_\theta(x_t^i, t \mid c) - v_\phi^{\text{CFG}}(x_t^i, t \mid c)\big\|_2^2\Big]. \tag{11}$$

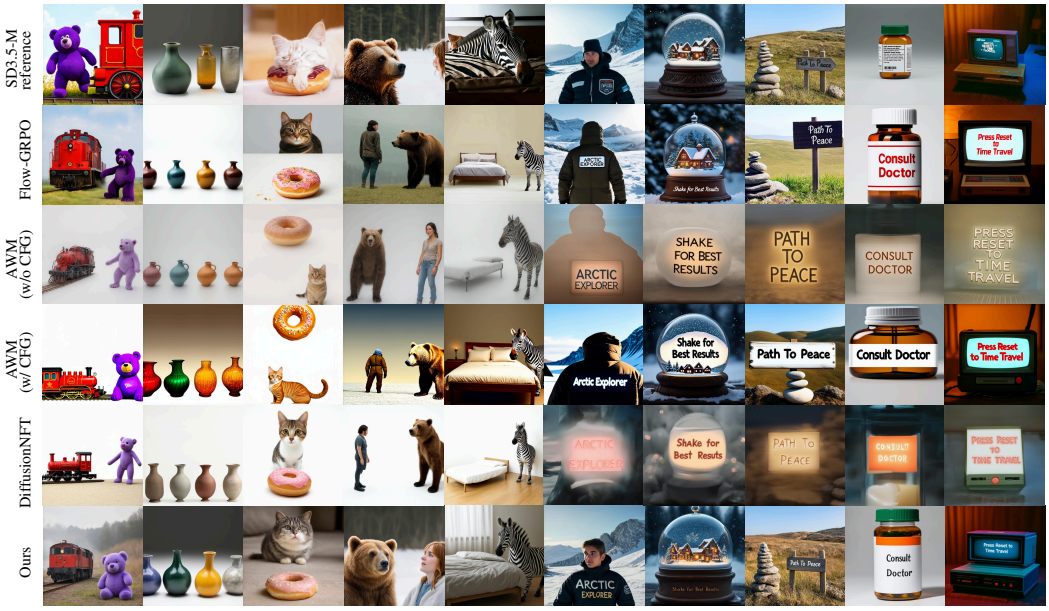


Figure 3: Visual comparison between baselines and our models.

Table 1: **Performance on Compositional Image Generation, Visual Text Rendering, and Human Preference benchmarks.** Task metrics are evaluated on the corresponding test prompts; image quality and preference scores are evaluated on `DrawBench` (Saharia et al., 2022) prompts. Baseline results are taken from Flow-GRPO (Liu et al., 2025b) or reimplemented when unavailable in original papers. `ImgRwd`: `ImageReward`.

Model	Task Metric		Image Quality	Preference Score			
	`GenEval` ↑	`OCR` ↑	`Aesthetics` ↑	`ImgRwd` ↑	`PickScore` ↑	`CLIPScore` ↑	`HPSv2.1` ↑
SD3.5-M	0.63	0.59	5.39	0.87	22.34	27.99	0.279
Compositional Image Generation							
Flow-GRPO (Liu et al., 2025b) (w/o KL)	0.95	—	4.93	0.44	21.16	—	—
Flow-GRPO (Liu et al., 2025b) (w/ KL)	0.95	—	5.25	1.03	22.37	29.25	0.274
AWM (Xue et al., 2025a) (w/o CFG)	0.91	—	5.10	0.39	21.76	26.99	0.235
AWM (Xue et al., 2025a) (w/ CFG)	0.86	—	5.25	1.06	22.12	28.86	0.278
DiffusionNFT (Zheng et al., 2025)	0.92	—	5.30	0.63	22.07	27.25	0.253
CRD	0.93	—	5.44	0.98	22.48	28.44	0.284
+ CFG sampling=1.5	0.93	—	5.42	1.05	22.55	28.80	0.289
+ CFG sampling=3.0	0.92	—	5.37	1.10	22.43	29.00	0.287
Visual Text Rendering							
Flow-GRPO (Liu et al., 2025b) (w/o KL)	—	0.93	5.13	0.58	21.79	—	—
Flow-GRPO (Liu et al., 2025b) (w/ KL)	—	0.92	5.32	0.95	22.44	28.86	0.282
AWM (Xue et al., 2025a) (w/o CFG)	—	0.97	5.22	-0.64	20.82	24.54	0.203
AWM (Xue et al., 2025a) (w/ CFG)	—	0.96	5.35	0.96	22.36	28.70	0.283
DiffusionNFT (Zheng et al., 2025)	—	0.97	4.89	-0.81	20.53	24.08	0.196
CRD	—	0.92	5.33	0.87	22.40	28.48	0.281
+ CFG sampling=1.5	—	0.92	5.31	0.97	22.50	28.77	0.287
+ CFG sampling=3.0	—	0.87	5.28	1.04	22.44	29.16	0.290

The overall training objective. The model v_θ is trained to minimize the combination of the aforementioned CRD loss and initial CFG KL regularization:

$$\mathcal{L}_{\text{CRD_KL}} = \mathcal{L}_{\text{CRD}} + \mathcal{L}_{\text{KL}}. \tag{12}$$

4 Experiments

4.1 Experiment Setup

Models and training configuration. All experiments fine-tune Stable Diffusion 3.5-Medium (SD3.5-M) (Esser et al., 2024), following Flow-GRPO (Liu et al., 2025b). Unless stated otherwise, images are generated at 512×512 resolution with a group size $K = 24$ for the main results and a group size $K = 6$ for all the ablations. We train LoRA adapters (Hu et al., 2022) with $r = 32$ and $\alpha = 64$, and adopt the remaining optimization and implementation details from DiffusionNFT (Zheng et al., 2025). All main experiments are run on up to four NVIDIA H100 GPUs and ablations on a single NVIDIA H100 GPU, each completing in under 36 hours.

Table 2: **`GenEval` Result.** Best scores are in blue, second-best in green. Results for baselines are taken from Flow-GRPO (Liu et al., 2025b) or the respective original papers, and reimplemented where unavailable. Obj.: Object; Attr.: Attribute.

Model	Overall	Single Obj.	Two Obj.	Counting	Colors	Position	Attr. Binding
Autoregressive Models							
Show-o (Xie et al., 2024)	0.53	0.95	0.52	0.49	0.82	0.11	0.28
Janus-Pro-7B (Chen et al., 2025e)	0.80	0.99	0.89	0.59	0.90	0.79	0.66
GPT-4o (Hurst et al., 2024)	0.84	0.99	0.92	0.85	0.92	0.75	0.61
Diffusion and Flow Matching Models							
SD-XL (Podell et al., 2023)	0.55	0.98	0.74	0.39	0.85	0.15	0.23
FLUX.1 Dev (Black Forest Labs, 2024)	0.66	0.98	0.81	0.74	0.79	0.22	0.45
SD3.5-M (Esser et al., 2024)	0.63	0.98	0.78	0.50	0.81	0.24	0.52
Flow + RL							
SD3.5-M+Flow-GRPO (Liu et al., 2025b)	0.95	1.00	0.99	0.95	0.92	0.99	0.86
SD3.5-M+AWM (Xue et al., 2025a) (w/o CFG)	0.91	1.00	0.98	0.95	0.77	0.78	0.65
SD3.5-M+DiffusionNFT (Zheng et al., 2025)	0.92	1.00	0.99	0.96	0.82	0.78	0.73
SD3.5-M+CRD	0.93	1.00	0.98	0.92	0.88	0.90	0.73

Datasets and reward models. We study two non-differentiable reward settings: (i) compositional image generation using `GenEval` (Ghosh et al., 2023) as the reward, and (ii) visual text rendering using an `OCR` reward (Chen et al., 2023). For both tasks, we use the same training/evaluation prompt splits and the corresponding reward models as in Flow-GRPO (Liu et al., 2025b). To assess generalization (reward hacking) and overall image quality beyond the task reward, we additionally evaluate on `DrawBench` (Saharia et al., 2022) prompts and report a suite of metrics, including `PickScore` (Kirstain et al., 2023), `CLIPScore` (Hessel et al., 2021), `HPSv2.1` (Wu et al., 2023), `Aesthetics` (Schuhmann, 2022), and `ImageReward` (Xu et al., 2023), capturing image quality, image-text alignment, and human preference.

Further experimental details are provided in Appendix C.

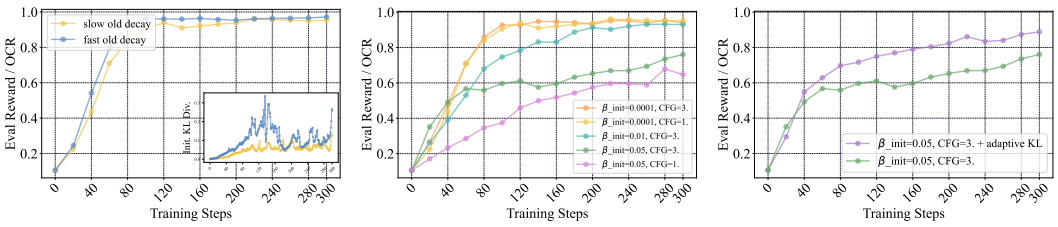
4.2 Main Results

Compositional image generation. In Tab. 1, CRD achieves a `GenEval` score of 0.93 (vs. 0.63 for SD3.5-M), and offers a strong quality–alignment trade-off. While Flow-GRPO (Liu et al., 2025b) reaches the highest `GenEval` (0.95), CRD attains the best `Aesthetics` (5.44) and strong preference metrics (`ImageReward`: 0.98, `PickScore`: 22.48, `HPSv2.1`: 0.284). Notably, CRD is the only method that improves `Aesthetics` over SD3.5-M (5.44 vs. 5.39) while also increasing preference scores. The detailed `GenEval` performance comparison is provided in Tab. 2.

Visual text rendering. CRD obtains 0.92 `OCR` accuracy, matching Flow-GRPO (w/ KL), while maintaining positive preference metrics on unseen prompts (`ImageReward`: 0.87, `PickScore`: 22.40, `HPSv2.1`: 0.281). In contrast, DiffusionNFT (Zheng et al., 2025) and AWM (Xue et al., 2025a) (w/o CFG) incur negative `ImageReward` (-0.81/-0.64), indicating degraded perceptual quality despite high `OCR` score.

Qualitative comparison. Figure 3 shows CRD preserves SD3.5-M photorealism while improving prompt fidelity, especially for multi-object composition and integrated text. Flow-GRPO often yields less natural backgrounds, AWM (w/ CFG) over-saturates toward a stylized look, and DiffusionNFT tends to produce uniform backgrounds or blurred regions around text. CRD produces more legible, naturally embedded text and coherent object relations, consistent with Tab. 1.

Global discussion on results. As shown in Tab. 1 and Fig. 3, although AWM does not require CFG during training, this setting leads to reward hacking and degraded visual quality; incorporating CFG partially alleviates this but introduces over-saturation artifacts. Although CRD implicitly distills the CFG guidance through Eq. (11), moderate test-time CFG further improves preference scores (Tab. 1), likely by increasing effective text-conditioning strength and due to the insufficient 'CFG distillation' in our RL fine-tuning. Based on the qualitative results in Fig. 3, we also notice that images generated with our method are closer to the reference model than the ones from the competing methods. Finally, although CRD underperforms Flow-GRPO on certain metrics, it inherits the efficiency of forward-process-based training and converges significantly faster, offering a more practical trade-off.



(a) Slow and fast old decay η_{old} (b) Results with different β_{init} (c) Ablation on adaptive $\hat{\beta}_{\text{init}}$

Figure 4: Ablations on slow old model decay rate η_{old}, and initial KL strength β_{init}.



Figure 5: Visual comparison corresponding to the ablations in Fig. 4.

Additional experimental results can be found in Appendix H.

4.3 Ablation Studies

We study the impact of three core design choices: the old-model decay rate η_{old}, the initial KL strength β_{init}, and the adaptive KL schedule $\hat{\beta}_{\text{init}}$. More ablations can be found in Appendix H.

Old-model decay (slow vs. fast). We first validate the proposed *slow* old-model decay. As shown in Fig. 4a, using a *fast* decay yields faster convergence, but it also causes highly unstable KL divergence with model ϕ. This instability and large KL value usually correspond to severe reward hacking, consistent with the qualitative results in Fig. 5: the fast-decay generations collapse to plain text on flat backgrounds, satisfying the `OCR` reward while losing all scene coherence.

Initial KL strength. In Fig. 4b, we report evaluation reward curves under different initial KL coefficients β_{init} and CFG values. We observe a clear trade-off between the strength of the initial KL constraint and training speed, where smaller β_{init} accelerates optimization. Moreover, the visual comparisons in Fig. 5 indicate that a larger initial KL typically produces better perceptual quality.

Adaptive KL. In Fig. 4c, we evaluate the proposed adaptive weighting for the initial KL coefficient. The adaptive strategy improves training progress while incurring minimal degradation in generation quality, as supported by both the reward curves and the qualitative results in Fig. 5.

5 Conclusion

We presented **Centered Reward Distillation (CRD)**, a forward-process diffusion RL framework grounded in KL-regularized reward maximization. Our key insight, that the intractable prompt-dependent normalizer cancels under within-prompt reward centering, yields a well-posed family of reward-matching objectives that unifies prior methods as special cases. Paired with practical techniques targeting distribution drift and reward hacking, CRD achieves competitive reward optimization with fast and stable training on `GenEval` and `OCR` benchmarks, and competitive results on unseen metrics.

AI USE STATEMENT

In this work, we used generative AI tools for implementing methods: AI coding assistants helped write and debug parts of the training and evaluation code used in our experiments. We have not used generative AI tools to generate synthetic datasets (the images used for RL fine-tuning are sampled from the diffusion model under study, as part of the method), to help develop theoretical models or conceptual frameworks, to formulate mathematical claims, to provide critical ingredients for proving mathematical claims, to assist in writing proofs, to propose or refine hypotheses, to design or provide feedback on the research methodology or experiments, to assist with translation, to clean or reformat datasets, or to interpret results; supporting qualitative and thematic data analysis is not applicable to this work. Additionally, we used generative AI tools to edit the paper for readability (correcting typos and grammar). We have reviewed all AI-assisted work: AI-assisted code was checked by the authors before being used in our experiments, and all AI-suggested text edits were verified by the authors, who take full responsibility for the content of this paper.

REPRODUCIBILITY STATEMENT

We provide the details needed to reproduce our results. The full training procedure of CRD is given in Algorithm 1, and Tab. 3 lists the hyperparameters and configurations for the `GenEval`, `OCR`, and multi-reward runs, including LoRA settings, rollout and evaluation sampling steps, group size, EMA schedules, KL strengths, and optimizer settings; the ELBO weighting variant used in all experiments is given in Eq. (13). Our implementation builds on the public DiffusionNFT codebase (Zheng et al., 2025), fine-tunes the publicly available SD3.5-M model (Esser et al., 2024), and, for the `GenEval` and `OCR` tasks, uses the same training/evaluation prompt splits and reward models as Flow-GRPO (Liu et al., 2025b); preference metrics are computed on `DrawBench` prompts with publicly available scorers (Sec. 4). Compute requirements are reported in Sec. 4, and Appendix C describes how we re-ran the AWM and DiffusionNFT baselines. Derivations supporting our theoretical claims are provided in Appendices D to G. Our code will be made publicly available upon acceptance.

REFERENCES

Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pp. 4447–4455. PMLR, 2024.

Ahmad Beirami, Alekh Agarwal, Jonathan Berant, Alexander D'Amour, Jacob Eisenstein, Chirag Nagpal, and Ananda Theertha Suresh. Theoretical guarantees on the best-of-n alignment policy. *arXiv preprint arXiv:2401.01879*, 2024.

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

Black Forest Labs. Flux. `https://blackforestlabs.ai/announcing-black-forest-labs/`, August 2024.

Luc Boudier, Loris Manganelli, Eleftherios Tsonis, Nicolas Dufour, and Vicky Kalogeiton. Training-free synthetic data generation with dual ip-adapter guidance. In *British Machine Vision Conference (BMVC)*, 2025.

Huayu Chen, Guande He, Lifan Yuan, Ganqu Cui, Hang Su, and Jun Zhu. Noise contrastive alignment of language models with explicit rewards. *Advances in Neural Information Processing Systems*, 37:117784–117812, 2024a.

Huayu Chen, Hang Su, Peize Sun, and Jun Zhu. Toward guidance-free ar visual generation via condition contrastive alignment. *arXiv preprint arXiv:2410.09347*, 2024b.

Huayu Chen, Kai Jiang, Kaiwen Zheng, Jianfei Chen, Hang Su, and Jun Zhu. Visual generation without guidance. *arXiv preprint arXiv:2501.15420*, 2025a.

Huayu Chen, Kaiwen Zheng, Qinsheng Zhang, Ganqu Cui, Yin Cui, Haotian Ye, Tsung-Yi Lin, Ming-Yu Liu, Jun Zhu, and Haoxiang Wang. Bridging supervised learning and reinforcement learning in math reasoning. *arXiv preprint arXiv:2505.18116*, 2025b.

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.

Kaijie Chen, Zhiyang Xu, Ying Shen, Zihao Lin, Yuguang Yao, and Lifu Huang. Superflow: Training flow matching models with rl on the fly. *arXiv preprint arXiv:2512.17951*, 2025c.

Renjie Chen, Wenfeng Lin, Yichen Zhang, Jiangchuan Wei, Boyuan Liu, Chao Feng, Jiao Ran, and Mingyu Guo. Towards self-improvement of diffusion models via group preference optimization. *arXiv preprint arXiv:2505.11070*, 2025d.

Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025e.

Jaemoo Choi, Yuchen Zhu, Wei Guo, Petr Molodyk, Bo Yuan, Jinbin Bai, Yi Xin, Molei Tao, and Yongxin Chen. Rethinking the design space of reinforcement learning for diffusion models: On the importance of likelihood estimation beyond loss design. *arXiv preprint arXiv:2602.04663*, 2026.

Geoffrey Cideron, Andrea Agostinelli, Johan Ferret, Sertan Girgin, Romuald Elie, Olivier Bachem, Sarah Perrin, and Alexandre Ramé. Diversity-rewarded cfg distillation. *arXiv preprint arXiv:2410.06084*, 2024.

Kevin Clark, Paul Vicol, Kevin Swersky, and David J Fleet. Directly fine-tuning diffusion models on differentiable rewards. *arXiv preprint arXiv:2309.17400*, 2023.

Robin Courant, Xi Wang, David Loiseaux, Marc Christie, and Vicky Kalogeiton. Pulp motion: Framing-aware multimodal camera and human motion generation. *arXiv preprint arXiv:2510.05097*, 2025.

Lucas Degeorge, Arijit Ghosh, Nicolas Dufour, David Picard, and Vicky Kalogeiton. How far can we go with imagenet for text-to-image generation? *arXiv*, 2025.

Haoyou Deng, Keyu Yan, Chaojie Mao, Xiang Wang, Yu Liu, Changxin Gao, and Nong Sang. Densegrpo: From sparse to dense reward for flow matching model alignment. *arXiv preprint arXiv:2601.20218*, 2026.

Zheng Ding and Weirui Ye. Treegrpo: Tree-advantage grpo for online rl post-training of diffusion models. *arXiv preprint arXiv:2512.08153*, 2025.

Nicolas Dufour, Victor Besnier, Vicky Kalogeiton, and David Picard. Don't drop your samples! coherence-aware training benefits conditional diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6264–6273, 2024.

Jacob Eisenstein, Chirag Nagpal, Alekh Agarwal, Ahmad Beirami, Alex D'Amour, DJ Dvijotham, Adam Fisch, Katherine Heller, Stephen Pfohl, Deepak Ramachandran, et al. Helping or herding? reward model ensembles mitigate but do not eliminate reward hacking. *arXiv preprint arXiv:2312.09244*, 2023.

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 *Forty-first International Conference on Machine Learning*, 2024.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.

Ying Fan and Kangwook Lee. Optimizing ddpm sampling with shortcut fine-tuning. *arXiv preprint arXiv:2301.13362*, 2023.

Adam Fisch, Jacob Eisenstein, Vicky Zayats, Alekh Agarwal, Ahmad Beirami, Chirag Nagpal, Pete Shaw, and Jonathan Berant. Robust preference optimization through reward model distillation. *arXiv preprint arXiv:2405.19316*, 2024.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pp. 10835–10866. PMLR, 2023.

Zhaolin Gao, Jonathan Chang, Wenhao Zhan, Owen Oertell, Gokul Swamy, Kianté Brantley, Thorsten Joachims, Drew Bagnell, Jason D Lee, and Wen Sun. Rebel: Reinforcement learning via regressing relative rewards. *Advances in Neural Information Processing Systems*, 37: 52354–52400, 2024.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36: 52132–52152, 2023.

Dongyoung Go, Tomasz Korbak, Germán Kruszewski, Jos Rozen, Nahyeon Ryu, and Marc Dymetman. Aligning language models with preferences through f-divergence minimization. *arXiv preprint arXiv:2302.08215*, 2023.

Alexey Gorbatovski, Boris Shaposhnikov, Alexey Malakhov, Nikita Surnachev, Yaroslav Aksenov, Ian Maksimov, Nikita Balagansky, and Daniil Gavrilov. Learn your reference model for real good alignment. *arXiv preprint arXiv:2404.09656*, 2024.

Lin Gui, Cristina Gârbacea, and Victor Veitch. Bonbon alignment for large language models and the sweetness of best-of-n sampling. *Advances in Neural Information Processing Systems*, 37: 2851–2885, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Dailan He, Guanlin Feng, Xingtong Ge, Yazhe Niu, Yi Zhang, Bingqi Ma, Guanglu Song, Yu Liu, and Hongsheng Li. Neighbor grpo: Contrastive ode policy optimization aligns flow models. *arXiv preprint arXiv:2511.16955*, 2025a.

Haoran He, Yuxiao Ye, Jie Liu, Jiajun Liang, Zhiyong Wang, Ziyang Yuan, Xintao Wang, Hangyu Mao, Pengfei Wan, and Ling Pan. Gardo: Reinforcing diffusion models without reward hacking. *arXiv preprint arXiv:2512.24138*, 2025b.

Xiaoxuan He, Siming Fu, Yuke Zhao, Wanli Li, Jian Yang, Dacheng Yin, Fengyun Rao, and Bo Zhang. Tempflow-grpo: When timing matters for grpo in flow models. *arXiv preprint arXiv:2508.04324*, 2025c.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*, 2021.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv preprint arXiv:2204.03458*, 2022.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 2022.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *arXiv preprint arXiv:2206.00364*, 2022.

Sunwoo Kim, Minkyu Kim, and Dongmin Park. Test-time alignment of diffusion models without reward over-optimization. *arXiv preprint arXiv:2501.05803*, 2025.

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. *Advances in neural information processing systems*, 36:36652–36663, 2023.

Tomasz Korbak, Hady Elsahar, Germán Kruszewski, and Marc Dymetman. On reinforcement learning and distribution matching for fine-tuning language models with no catastrophic forgetting. *Advances in Neural Information Processing Systems*, 35:16203–16220, 2022.

Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. *Advances in Neural Information Processing Systems*, 37:122458–122483, 2024.

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.

Yuming Li, Yikai Wang, Yuying Zhu, Zhongyu Zhao, Ming Lu, Qi She, and Shanghang Zhang. Branchgrpo: Stable and efficient grpo with structured branching in diffusion models. *arXiv preprint arXiv:2509.06040*, 2025.

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

Henglin Liu, Huijuan Huang, Jing Wang, Chang Liu, Xiu Li, and Xiangyang Ji. Diverse-grpo: Mitigating mode collapse in image generation via diversity-aware grpo. *arXiv preprint arXiv:2512.21514*, 2025a.

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. *arXiv preprint arXiv:2505.05470*, 2025b.

Qiang Liu. Icml tutorial on the blessing of flow. *International conference on machine learning*, 2025.

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.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025c.

Yifu Luo, Penghui Du, Bo Li, Sinan Du, Tiantian Zhang, Yongzhe Chang, Kai Wu, Kun Gai, and Xueqian Wang. Sample by step, optimize by chunk: Chunk-level grpo for text-to-image generation. *arXiv preprint arXiv:2510.21583*, 2025a.

Yihong Luo, Tianyang Hu, and Jing Tang. Reinforcing diffusion models by direct group preference optimization. *arXiv preprint arXiv:2510.08425*, 2025b.

Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, et al. Inference-time scaling for diffusion models beyond scaling denoising steps. *arXiv preprint arXiv:2501.09732*, 2025a.

Yuhang Ma, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. Hpsv3: Towards wide-spectrum human preference score. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15086–15095, 2025b.

Weijia Mao, Hao Chen, Zhenheng Yang, and Mike Zheng Shou. The image as its own reward: Reinforcement learning with adversarial reward for image generation. *arXiv preprint arXiv:2511.20256*, 2025.

Xin Mao, Feng-Lin Li, Huimin Xu, Wei Zhang, and Anh Tuan Luu. Don't forget your reward values: Language model alignment via value-based calibration. *arXiv preprint arXiv:2402.16030*, 2024.

Simon Matrenok, Skander Moalla, and Caglar Gulcehre. Quantile reward policy optimization: Alignment with pointwise regression and exact partition functions. *arXiv preprint arXiv:2507.08068*, 2025.

David McAllister, Songwei Ge, Brent Yi, Chung Min Kim, Ethan Weber, Hongsuk Choi, Haiwen Feng, and Angjoo Kanazawa. Flow matching policy gradients. *arXiv preprint arXiv:2507.21053*, 2025.

Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14297–14306, 2023.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Zijing Ou, Jacob Si, Junyi Zhu, Ondrej Bohdal, Mete Ozay, Taha Ceritli, and Yingzhen Li. Diffusion alignment beyond kl: Variance minimisation as effective policy optimiser. *arXiv preprint arXiv:2602.12229*, 2026.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

Jean Pachebat, Giovanni Conforti, Alain Durmus, and Yazid Janati. Iterative tilting for diffusion fine-tuning. *arXiv preprint arXiv:2512.03234*, 2025.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

Jan Peters and Stefan Schaal. Reinforcement learning by reward-weighted regression for operational space control. In *Proceedings of the 24th international conference on Machine learning*, pp. 745–750, 2007.

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. *arXiv preprint arXiv:2307.01952*, 2023.

Peter Potaptchik, Cheuk-Kit Lee, and Michael S Albergo. Tilt matching for scalable sampling and fine-tuning. *arXiv preprint arXiv:2512.21829*, 2025.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Allen Z Ren, Justin Lidard, Lars L Ankile, Anthony Simeonov, Pulkit Agrawal, Anirudha Majumdar, Benjamin Burchfiel, Hongkai Dai, and Max Simchowitz. Diffusion policy policy optimization. *arXiv preprint arXiv:2409.00588*, 2024.

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*, pp. 10684–10695, 2022.

Amirmojtaba Sabour, Michael S Albergo, Carles Domingo-Enrich, Nicholas M Boffi, Sanja Fidler, Karsten Kreis, and Eric Vanden-Eijnden. Test-time scaling of diffusions with flow maps. *arXiv preprint arXiv:2511.22688*, 2025.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

Christoph Schuhmann. Laion-aesthetics. `https://laion.ai/blog/laion-aesthetics/`, 2022.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pp. 2256–2265. PMLR, 2015.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2020.

Masatoshi Uehara, Yulai Zhao, Chenyu Wang, Xiner Li, Aviv Regev, Sergey Levine, and Tommaso Biancalani. Inference-time alignment in diffusion models with reward-guided generation: Tutorial and review. *arXiv preprint arXiv:2501.09685*, 2025.

Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *Advances in Neural Information Processing Systems*, 34:11287–11302, 2021.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8228–8238, 2024.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Feng Wang and Zihao Yu. Coefficients-preserving sampling for reinforcement learning with flow matching. *arXiv preprint arXiv:2509.05952*, 2025.

Fu-Yun Wang, Han Zhang, Michael Gharbi, Hongsheng Li, and Taesung Park. Promptrl: Prompt matters in rl for flow-based image generation. *arXiv preprint arXiv:2602.01382*, 2026a.

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. *arXiv preprint arXiv:2510.22319*, 2025a.

Xi Wang, Nicolas Dufour, Nefeli Andreou, Marie-Paule Cani, Victoria Fernández Abrevaya, David Picard, and Vicky Kalogeiton. Analysis of classifier-free guidance weight schedulers. *arXiv preprint arXiv:2404.13040*, 2024.

Xi Wang, Robin Courant, Marc Christie, and Vicky Kalogeiton. Akira: Augmentation kit on rays for optical video generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 2609–2619, 2025b.

Yiyang Wang, Xi Chen, Xiaogang Xu, Yu Liu, and Hengshuang Zhao. Gdro: Group-level reward post-training suitable for diffusion models. *arXiv preprint arXiv:2601.02036*, 2026b.

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.

Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528*, 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. *Advances in Neural Information Processing Systems*, 36:15903–15935, 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.

Haotian Ye, Kaiwen Zheng, Jiashu Xu, Puheng Li, Huayu Chen, Jiaqi Han, Sheng Liu, Qinsheng Zhang, Hanzi Mao, Zekun Hao, et al. Data-regularized reinforcement learning for diffusion models at scale. *arXiv preprint arXiv:2512.04332*, 2025.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6613–6623, 2024.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Huizhuo Yuan, Zixiang Chen, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning of diffusion models for text-to-image generation. *Advances in Neural Information Processing Systems*, 37: 73366–73398, 2024.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025.

Kaichen Zhang, Yuzhong Hong, Junwei Bao, Hongfei Jiang, Yang Song, Dingqian Hong, and Hui Xiong. Gvpo: Group variance policy optimization for large language model post-training. *arXiv preprint arXiv:2504.19599*, 2025.

Shengjun Zhang, Zhang Zhang, Chensheng Dai, and Yueqi Duan. E-grpo: High entropy steps drive effective reinforcement learning for flow models. *arXiv preprint arXiv:2601.00423*, 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. *arXiv preprint arXiv:2509.16117*, 2025.

Yujie Zhou, Pengyang Ling, Jiazi Bu, Yibin Wang, Yuhang Zang, Jiaqi Wang, Li Niu, and Guangtao Zhai. Fine-grained grpo for precise preference alignment in flow models. *arXiv preprint arXiv:2510.01982*, 2025.

Xuekai Zhu, Daixuan Cheng, Dinghuai Zhang, Hengli Li, Kaiyan Zhang, Che Jiang, Youbang Sun, Ermo Hua, Yuxin Zuo, Xingtai Lv, et al. Flowrl: Matching reward distributions for llm reasoning. *arXiv preprint arXiv:2509.15207*, 2025a.

Yuchen Zhu, Wei Guo, Jaemoo Choi, Petr Molodyk, Bo Yuan, Molei Tao, and Yongxin Chen. Enhancing reasoning for diffusion llms via distribution matching policy optimization. *arXiv preprint arXiv:2510.08233*, 2025b.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

Appendix for CRD

This appendix is organized as follows:

- Appendix A: Limitation and future work.
- Appendix B: More discussion on related works.
- Appendix C: Experimental details.
- Appendix D: Theoretical discussion on DiffusionNFT.
- Appendix E: Discussion of accumulated tilting.
- Appendix F: GVPO and Reward Distill as special cases of CRD.
- Appendix G: Ratio-based Reward Distillation and connection to InfoNCA.
- Appendix H: Additional experimental results and visualizations.

A Limitations and Future Work

Despite strong empirical performance, this work has several limitations that suggest clear directions for future research. *First*, while CRD is motivated by a KL-regularized optimality view, a more rigorous theoretical analysis remains open including convergence and stability guarantees under the practical approximations such as ELBO-based log-ratio surrogates, moving references, and finite-sample within-prompt centering. *Second*, CRD is fundamentally constrained by the coverage and capabilities of the reference (pretrained) model, since training relies on self-generated samples: if the base model rarely produces trajectories that exhibit a desired behavior, the reward signal provides little leverage to bootstrap it, and continued self-training can further narrow diversity. A natural mitigation is to inject real data and explicit supervision into the training pipeline (Chen et al., 2025b). For example, mixing RL updates with Supervised Fine-Tuning (SFT) updates or adding a data-anchoring regularizer (data KL) to preserve coverage and diversity (Ye et al., 2025). *Finally*, although CRD is substantially more efficient than Flow-GRPO variants, it still incurs non-trivial computational overhead relative to standard fine-tuning as it requires full sampling during training. This cost becomes more pronounced for video diffusion models with long temporal trajectories and high-resolution frames (Wan et al., 2025). Future work could reduce training costs through integration with faster samplers or diffusion distillation techniques to amortize generation costs while maintaining reward optimization performance.

B More Discussion with Related Works

This section extends the discussion of related works in Sec. 2.

GRPO in visual generation. Recently, methods leveraging *Group Relative Policy Optimization* (GRPO) (Shao et al., 2024; Guo et al., 2025) have shown strong alignment performance in diffusion and flow-based models (Liu et al., 2025b; Xue et al., 2025b). Meanwhile, a recurring challenge in diffusion/flow GRPO is the denoising process itself: learning signals and estimator variance are highly timestep-dependent, while practical training also hinges on rollout fidelity, sampling efficiency, and the choice of reward objective. Existing methods address these issues through several complementary mechanisms:

1. **Trajectory structuring for exploration and credit assignment.** These methods modify rollout topology (branching/trees/chunks) to broaden exploration and improve credit assignment along denoising: TempFlow-GRPO (He et al., 2025c); Branch-GRPO (Li et al., 2025), TreeGRPO (Ding & Ye, 2025); Chunk-GRPO (Luo et al., 2025a). The trade-off is added algorithmic complexity and sensitivity to design choices (e.g., branching depth, chunk size).
2. **Dense rewards and ratio/regularization stabilizers.** This line reshapes timestep-wise learning signals and stabilizes per-step weighting via dense rewards or ratio/regularization

control: Dense-GRPO (Deng et al., 2026), GARDO (He et al., 2025b), GRPO-Guard (RatioNorm) (Wang et al., 2025a). Gains depend on the quality of dense reward proxies and careful calibration of normalization/regularization.

3. **Sampling efficiency and rollout fidelity.** These works improve the compute–quality trade-off by increasing group-sampling efficiency and reducing solver-induced artifacts: SuperFlow (Chen et al., 2025c), Flow-CPS (Wang & Yu, 2025), E-GRPO (Zhang et al., 2026), Neighbor GRPO (He et al., 2025a). They are largely orthogonal to reward design but can remain brittle under sparse or misaligned rewards.
4. **Richer reward definitions and objectives.** Recent variants broaden supervision beyond a single prompt-level scalar (e.g., reference-image rewards, diversity objectives, prompt refinement, or multi-granularity advantages): Adv-GRPO (Mao et al., 2025), DiverseGRPO (Liu et al., 2025a), PromptRL (Wang et al., 2026a), Granular-GRPO (Zhou et al., 2025). While often more informative, they introduce extra assumptions/modules and can increase sensitivity to reward/model misspecification.

Despite these advances, a fundamental limitation remains: most of these methods optimize through the *backward-process* MDP formulation, introducing inherent challenges in timestep-dependent variance, rollout overhead, and credit assignment that ultimately limit training efficiency.

Diffusion RL based on the forward process. In contrast to Flow-GRPO and subsequent methods that explicitly model the diffusion sampling trajectory as an MDP and derive policy-gradient style updates under that assumption, a recent line of work argues that diffusion RL can be formulated more directly using the *forward-process* (Xue et al., 2025a; Zheng et al., 2025; Luo et al., 2025b; Wang et al., 2026b; Choi et al., 2026; Chen et al., 2025d; Zhu et al., 2025b). The key idea is to decouple data sampling and model training: in the sampling phase, the model generates clean samples without storing intermediate noisy states; in the training phase, the loss is computed on noisy versions of the samples using the forward diffusion process. Experiments and analyses from these works suggest that forward-process-based methods achieve significantly faster training compared to backward-process-based methods like Flow-GRPO.

GPO (Chen et al., 2025d) is the first attempt at this perspective. It employs an ELBO-based likelihood estimator for diffusion models and shows that the resulting objective can be interpreted as an *advantage-weighted log-likelihood*, connecting diffusion RL to weighted maximum likelihood updates. Concurrently, AWM (Xue et al., 2025a) analyzes the variance properties of Flow-GRPO-style estimators and proposes a forward-process algorithm, *Advantage Weighted Matching*, to mitigate variance amplification, demonstrating that it significantly accelerates training. An important conceptual link is that the GPO objective can be viewed as a first-order approximation of AWM objective; the approximation error becomes negligible when the current model remains close to the reference (old) model (Xue et al., 2025a; Chen et al., 2025d).

DiffusionNFT (Zheng et al., 2025) also adopts a forward-process RL formulation and advocates using faster ODE-based samplers during training to improve efficiency. Unlike approaches that explicitly rely on an ELBO likelihood estimator to construct loss function, DiffusionNFT uniquely derives an update that targets an optimal velocity direction that steers samples toward high-reward regions. Despite this difference in derivation, DiffusionNFT remains conceptually aligned with the forward-process family in that it targets the same optimal solution.

DGPO (Luo et al., 2025b) further develops the forward-process paradigm by learning from *group-level* preferences, exploiting relative information among samples within the same group and empirically demonstrating efficient optimization and stable convergence. GDRO (Wang et al., 2026b) reframes the log-likelihood ratio and advantage in terms of probability and derives a cross-entropy training objective; the resulting loss is closely related to InfoNCA proposed in Chen et al. (2024a). Finally, Choi et al. (2026) provides a systematic study of design choices in diffusion RL and empirically validates the central contribution from the ELBO-based estimator in diffusion RL training.

Relation to GVPO and VMPO. The objective of prior work GVPO (Zhang et al., 2025) can be viewed as a special case of our CRD. GVPO focuses on the variance analysis of the loss gradient and is primarily designed for LLM post-training. While the two methods share conceptual similarities, CRD is derived from a different perspective and is tailored for image generation tasks with text-to-image diffusion models. Furthermore, we demonstrate that CRD provides a unified framework

Algorithm 1 CRD Distillation

Require: Pre-trained model ϕ, reward model r_ψ, ODE solver $\mathrm{ODE}[\cdot]$, prompt dataset $\mathcal{D}_c$, β_{old} and β_{init}, group size K, EMA decay η_{old} and η_{samp}, temperature τ

$\Delta\theta \leftarrow \text{initLoRA}(),\ \Delta\theta_{\text{old}} \leftarrow \text{initLoRA}(),\ \Delta\theta_{\text{samp}} \leftarrow \text{initLoRA}()$ // init adapters

$\theta \triangleq \phi + \Delta\theta,\ \theta_{\text{old}} \triangleq \phi + \Delta\theta_{\text{old}},\ \theta_{\text{samp}} \triangleq \phi + \Delta\theta_{\text{samp}}$ // effective weights

repeat

  `### Sample prompts and group of data`

  Sample $c \sim \mathcal{D}_c$, $\{x_1^i\}_{i=1}^K \sim \mathcal{N}(0,1)$

  Generate a group of samples $\{x_0^i\}_{i=1}^K \leftarrow \mathrm{ODE}[v_{\theta_{\text{samp}}}](\{x_1^i\}_{i=1}^K)$

  `### Calculate r(c,x)`

  $r_{\text{raw}}^i = r_\psi(c, x_0^i)$ // scaled to $[0,1]$

  $r(c, x_0^i) = \frac{r_{\text{raw}}^i - \text{mean}(\{r_{\text{raw}}\}^{1:K})}{\text{std}(\{r_{\text{raw}}\}^{1:K})}$

  `### Estimate` $R_\theta(c,x) \triangleq \beta \log \frac{p_\theta(x|c)}{p_{\text{old}}(x|c)}$

  Sample $t \sim \mathcal{U}[0,1]$ and noise $\epsilon \sim \mathcal{N}(0,1)$ for each i

  Calculate $x_t = t\epsilon + (1-t)x_0$ and $v_{\text{target}} = \epsilon - x_0$ for each i

  $R_\theta(c, x_0^i) \leftarrow -\beta_{\text{old}}(||v_\theta - v_{\text{target}}||^2 - ||v_{\theta_{\text{old}}} - v_{\text{target}}||^2)$ for each i // ELBO estimator

  `### Loss calculation and update` θ

  Calculate $\Delta_{r,w}^i$ and $\Delta_{\widehat{R},w}^i$ using $r(c, x_0^i)$ and $\widehat{R}_\theta(c, x_0^i)$ // Eq. (6)

  Calculate $\mathcal{L}_{\text{CRD}}^{(\tau)} = \frac{1}{K}\sum_{i=1}^K \left(\Delta_{r,w}^i - \Delta_{\widehat{R},w}^i\right)^2$

  $\hat{\beta}_{\text{init}}^i \leftarrow r_{\text{raw}}^i \beta_{\text{init}}$ // Adaptive reference KL

  Calculate $\mathcal{L}_{\text{KL}} = \frac{1}{K}\sum_{i=1}^K \hat{\beta}_{\text{init}}^i ||v_\theta(x_t^i, t, c) - v_\phi^{\text{CFG}}(x_t^i, t, c)||^2$

  Update θ using gradient of $\mathcal{L}_{\text{CRD_KL}} = \mathcal{L}_{\text{CRD}} + \mathcal{L}_{\text{KL}}$

  `### Update` θ_{old} `and` θ_{samp}

  $\theta_{\text{old}} \leftarrow \eta_{\text{old}}\theta_{\text{old}} + (1-\eta_{\text{old}})\theta$

  $\theta_{\text{samp}} \leftarrow \eta_{\text{samp}}\theta_{\text{samp}} + (1-\eta_{\text{samp}})\theta$

until *convergence*

Return Reward tilted model θ

that subsumes both GVPO and prior two-sample reward distillation methods (Mao et al., 2024; Gao et al., 2024; Fisch et al., 2024).

More recently, Ou et al. (2026) introduce Variance Minimisation Policy Optimisation (VMPO), which formulates diffusion alignment as minimising the variance of log importance weights rather than directly optimising a KL-based objective. Despite the surface-level similarity in the final objective, VMPO is more closely related to a diffusion-adapted variant of GVPO (Zhang et al., 2025), accompanied by corresponding theoretical analysis. Compared to CRD, VMPO adopts a different empirical focus, and at the time of our submission, its scope remains narrower in terms of the experimental coverage that CRD provides.

Finally, for diffusion RL, all forward-process-based methods address different aspects of a broader picture. Together, they form a complementary recipe that enables faster training while mitigating reward hacking.

C Experimental Details

The full training procedure of CRD is summarized in Algorithm 1.

CRD is implemented on top of the DiffusionNFT codebase, with hyperparameters and model configurations listed in Tab. 3. Regarding training time, we compare with AWM and DiffusionNFT under the same training budget; however, as both baselines suffer from reward hacking, we report their results at early checkpoints where this issue is less severe. Throughout the experiments, we adopt the adaptive weighting (Yin et al., 2024; Zheng et al., 2025) variant of the implicit model reward:

$$\widehat{R}_\theta^{\text{adap}}(c,x) \triangleq -\beta\, \mathbb{E}_{t,\epsilon}\left[\frac{d\,\|v_\theta(x_t, t \mid c) - v_{\text{target}}\|_2^2}{\text{sg}(\|v_\theta(x_t, t \mid c) - v_{\text{target}}\|_1)} - \frac{d\,\|v_{\text{ref}}(x_t, t \mid c) - v_{\text{target}}\|_2^2}{\text{sg}(\|v_{\text{ref}}(x_t, t \mid c) - v_{\text{target}}\|_1)}\right], \quad (13)$$

where d is the dimension of x and $\text{sg}(\cdot)$ denotes the stop-gradient operation. In practice, we apply stop gradient to the weighted implicit reward $\sum_{j=1}^K w_j\, \widehat{R}_\theta(c, x_j)$ in Eq. (6).

Table 3: Hyperparameters and configurations used for main experiment runs. Model architecture is the same as the pretrained model. Largest Experiments in this work are conducted with 4 NVIDIA H100 GPUs. i in η_{old} and η_{samp} denotes for *Training (Optimizer) Steps*.

	Reward Model	OCR	GenEval	Multi Rewards
LoRA	α	64	64	64
	r	32	32	32
Sampling	Sampling CFG	1.0	1.0	1.0
	Training Rollout Steps	10	10	15
	Evaluation Sampling Steps	40	40	40
	ODE Solver	'dpmv2'	'dpmv2'	'dpmv2'
Forward-Process RL	Generation Image Resolution	512×512	512×512	512×512
	Group Size K	24	24	24
	Training (Optimizer) Steps	720	360	300
	Num of Groups per Batch	48	48	48
	Optimizer Step per Batch	2	1	1
CRD	β_{old}	1.0	1.0	0.1
	β_{init}	0.1	0.08	0.01
	CFG for Initial KL	4.5	4.5	4.5
	Use Adaptive KL Strength $\hat{\beta}_{\text{init}}$	True	True	True
	η_{old}	$\min(0.25 + 0.005i,\ 0.999)$	$\min(0.1 + 0.001i,\ 0.5)$	$\min(0.5 + 0.0025i,\ 0.999)$
	η_{samp}	$\min(\max(0.0075(i-75),\ 0),\ 0.999)$	$\min(0.001i,\ 0.5)$	$\min(0.001i,\ 0.5)$
Optimization	Optimizer	AdamW	AdamW	AdamW
	(β_1, β_2)	(0.9,0.999)	(0.9,0.999)	(0.9,0.999)
	Learning Rate	3e-4	3e-4	3e-4
	Weight Decay	1e-4	1e-4	1e-4
	EMA Ratio	0.9	0.9	0.9

D DiffusionNFT Recovers the KL-Regularized Exponential-Tilt Optimum

Unlike other formulations that explicitly include the log density ratio between the current and reference models and rely on a diffusion ELBO estimator to approximate it (an approach that generalizes naturally to LLMs), DiffusionNFT is derived directly for diffusion and flow models without invoking the log density ratio, making it a 'native' diffusion RL algorithm. Nevertheless, we show in this section that when the reference model is held fixed, DiffusionNFT is equivalent to optimizing the same *exponentially tilted* target distribution as standard KL-regularized reward maximization (see Eq. (1)), revealing a deep connection between DiffusionNFT and our method. We present the argument at the density level using the notation of the main text.

D.1 DiffusionNFT Score Update Implies an Exponential Tilt

DiffusionNFT is defined through a modification of the velocity (or score) field of the diffusion model relative to the old model. Let $p^*(\cdot \mid c)$ denote the population-optimal distribution induced by the optimal velocity field v^* for a *fixed* old/reference model p_{old}. Under the fixed Gaussian noising family, for two diffusion processes A and B, the differences of velocity fields reduce to differences of score functions:

$$v^A(x_t, t \mid c) - v^B(x_t, t \mid c) \;=\; \kappa(t)\Big(\nabla_{x_t} \log p_t^A(x_t \mid c) - \nabla_{x_t} \log p_t^B(x_t \mid c)\Big), \tag{14}$$

where $\kappa(t)$ depends only on the noise schedule, and $p_t^A(\cdot \mid c)$ and $p_t^B(\cdot \mid c)$ denote the noised marginals at time t induced by diffusion process A and B, respectively.

DiffusionNFT constructs a guidance direction of the form

$$\Delta(x_t, c, t) \;=\; \alpha(x_t, c)\big(v^+(x_t, t \mid c) - v^{\text{old}}(x_t, t \mid c)\big), \tag{15}$$

where $\alpha(x_t, c)$ is an "optimality" score (in DiffusionNFT, an optimality posterior). Using the Bayes relation

$$p_t^+(x_t \mid c) \;=\; p_t^{\text{old}}(x_t \mid c)\, \frac{\alpha(x_t, c)}{\mathbb{E}_{x_t \sim p_t^{\text{old}}(\cdot \mid c)}[\alpha(x_t, c)]}, \tag{16}$$

and applying Eq. (14), one obtains

$$v^+(x_t, t \mid c) - v^{\text{old}}(x_t, t \mid c) \;=\; \kappa(t) \nabla_{x_t} \log \alpha(x_t, c), \tag{17}$$

and therefore

$$\Delta(x_t, c, t) \;=\; \kappa(t)\, \alpha(x_t, c)\, \nabla_{x_t} \log \alpha(x_t, c) \;=\; \kappa(t)\, \nabla_{x_t} \alpha(x_t, c). \tag{18}$$

In the population optimum of DiffusionNFT (cf. the optimality condition in the main text), the optimal velocity satisfies (see Eq. (3) in DiffusionNFT)

$$v^*(x_t, t \mid c) - v^{\text{old}}(x_t, t \mid c) \;=\; \frac{2}{\beta}\,\Delta(x_t, c, t). \tag{19}$$

Combining Eq. (14), Eq. (18), and Eq. (19) yields a score-level characterization:

$$\nabla_{x_t} \log \frac{p_t^*(x_t \mid c)}{p_t^{\text{old}}(x_t \mid c)} \;=\; \frac{2}{\beta}\,\nabla_{x_t}\alpha(x_t, c). \tag{20}$$

Integrating both sides over x_t gives the corresponding density-level optimum:

$$p_t^*(x_t \mid c) \;=\; \frac{1}{Z_t(c)}\; p_t^{\text{old}}(x_t \mid c)\; \exp\!\left(\frac{2}{\beta}\,\alpha(x_t, c)\right), \tag{21}$$

where $Z_t(c) = \int p_t^{\text{old}}(x_t \mid c) \exp\!\left(\frac{2}{\beta}\alpha(x_t, c)\right)\, dx_t$.

D.2 Reduction to KL-regularized RL at $t = 0$

At the data level ($t = 0$), we identify $\alpha(x_0, c)$ with the (scaled) reward used for training; in our notation we take

$$\alpha(x_0, c) \equiv r(c, x_0). \tag{22}$$

Substituting Eq. (22) into Eq. (21) at $t = 0$ yields

$$p^*(x_0 \mid c) \;\propto\; p^{\text{old}}(x_0 \mid c) \exp\!\left(\frac{2}{\beta}\, r(c, x_0)\right). \tag{23}$$

Eq. (23) is exactly the KL-regularized exponential-tilt optimum Eq. (1), with reference $p_{\text{ref}} = p_{\text{old}}$ and an effective inverse-temperature $\lambda = 2/\beta$. Therefore, *with a fixed reference model*, DiffusionNFT is equivalent (at the population optimum) to KL-regularized reward maximization targeting the same exponentially tilted distribution (see Eqs. (1) and (23)).

Implication. This equivalence makes explicit that DiffusionNFT performs KL-regularized policy improvement relative to p_{old}, and also clarifies why repeatedly updating the reference online can accumulate tilt (see Appendix E).

E Online RL Leads to Reward Hacking via Accumulated Tilting

This section explains why running KL-regularized diffusion fine-tuning *online* with a moving reference can induce *accumulated tilting* (Peters & Schaal, 2007; Kim et al., 2025), leading to increasingly peaked policies and potential reward hacking (see Figs. 3 and 5).

Online training with a moving reference. In practice, methods such as DiffusionNFT can be run online: after optimizing the student model p_θ for one epoch, the reference model is updated (e.g., by a hard copy of weights or by an Exponential Moving Average (EMA)), and the next epoch optimizes against this updated reference. This induces a recursion over the reference distributions across epochs.

E.1 Idealized Recursion under Exact Per-epoch Optima

Let $p^{(n)}(x \mid c)$ denote the reference ("old") model distribution at epoch n. We idealize the analysis by assuming that (i) the reward function $r(c, x)$ is fixed across epochs and (ii) at each epoch we reach the population optimum of the KL-regularized objective with inverse-temperature $\beta > 0$. Under this idealization, the population-optimal model at epoch $n + 1$ is the exponentially tilted version of the reference:

$$p^{(n+1)}(x \mid c) \;=\; \frac{1}{Z^{(n)}(c)}\; p^{(n)}(x \mid c)\; \exp\!\left(\frac{1}{\beta}\, r(c, x)\right), \tag{24}$$

where $Z^{(n)}(c) = \int p^{(n)}(x \mid c) \exp(r(c, x)/\beta)\, dx$ is the normalizer. Eq. (24) is the same Boltzmann form as in Eq. (1), applied epoch-by-epoch with the updated reference.

Unrolling the recursion yields

$$p^{(N)}(x \mid c) \;\propto\; p^{(0)}(x \mid c)\; \exp\!\left(\frac{N}{\beta}\, r(c,x)\right). \tag{25}$$

Thus, under exact per-epoch optimization and a fixed reward $r(c,x)$, online training compounds the reward tilt: the effective inverse temperature scales linearly with the number of epochs N.

E.2 Concentration and Reward Hacking

Eq. (25) implies that as $N \to \infty$, the distribution $p^{(N)}(\cdot \mid c)$ concentrates its mass on the set of maximizers of the reward for that prompt. Let $r_{\max}(c) = \sup_x r(c,x)$ and define the maximizer set $S(c) = \{x : \; r(c,x) = r_{\max}(c)\}$. Under mild regularity assumptions (e.g., $p^{(0)}(\cdot \mid c)$ assigns nonzero mass near $S(c)$), the sequence $\{p^{(N)}(\cdot \mid c)\}$ becomes increasingly peaked on $S(c)$. In practice, when $r(c,x)$ is produced by an imperfect reward model, this progressive sharpening can amplify reward exploitation, i.e., *reward hacking*.

E.3 Remarks on EMA References

If the reference model is updated via EMA in parameter space, the induced recursion over distributions is not exactly the multiplicative update in Eq. (24). Nevertheless, EMA typically interpolates between keeping the reference fixed and fully replacing it with the current student, effectively reducing the rate at which the tilt coefficient grows with epoch. The qualitative conclusion remains: without an explicit anchoring term to the initial reference, repeated online updates tend to accumulate reward tilt and can lead to overly concentrated policies that are more prone to reward hacking.

F GVPO and Reward Distill as Special Cases

Extreme case: $K = 2$ and $\tau \to 0$ recovers reward distillation. Consider two samples $\{x^1, x^2\}$ with rewards $r_1 = r(c,x^1)$ and $r_2 = r(c,x^2)$, and assume w.l.o.g. $r_1 \geq r_2$. As $\tau \to 0$, the softmax becomes one-hot:

$$w_1 \to 1, \qquad w_2 \to 0.$$

Hence $\overline{r}_w \to r_1$ and $\overline{R_\theta}_w \to R_1$ where $R_i = R_\theta(c,x^i)$. The centered terms satisfy

$$\Delta^1_{r,w} \to 0, \quad \Delta^2_{r,w} \to r_2 - r_1, \qquad \Delta^1_{R,w} \to 0, \quad \Delta^2_{R,w} \to R_2 - R_1.$$

Plugging into Eq. (7) (uniformly averaging over $i \in \{1,2\}$) yields

$$\begin{aligned}
\mathcal{L}^{(\tau\to 0)}_{\text{distill_GVPO}} &= \mathbb{E}_{\rho(c,x^1,x^2)}\left[\frac{1}{2}\Big((0-0)^2 + \big((r_2 - r_1) - (R_2 - R_1)\big)^2\Big)\right] \\
&= \mathbb{E}_{\rho(c,x^1,x^2)}\left[\frac{1}{2}\Big((r_1 - r_2) - (R_1 - R_2)\Big)^2\right].
\end{aligned}$$

This is exactly the reward distillation loss proposed in Mao et al. (2024); Gao et al. (2024); Fisch et al. (2024).

Extreme case: $\tau \to \infty$ recovers GVPO (Zhang et al., 2025). When $\tau \to \infty$, the softmax weights become uniform:

$$w_i(c;\tau) \;=\; \frac{\exp(r(c,x_i)/\tau)}{\sum_{j=1}^K \exp(r(c,x_j)/\tau)} \;\xrightarrow[\tau\to\infty]{}\; \frac{1}{K}. \tag{26}$$

Therefore the weighted means reduce to simple averages,

$$\overline{r}_w(c,\{x_i\}) \xrightarrow[\tau\to\infty]{} \frac{1}{K}\sum_{j=1}^K r(c,x_j), \qquad \overline{R_\theta}_w(c,\{x_i\}) \xrightarrow[\tau\to\infty]{} \frac{1}{K}\sum_{j=1}^K R_\theta(c,x_j), \tag{27}$$

and consequently we show that GVPO is recovered as a special case of the proposed framework.

G RATIO-BASED REWARD DISTILLATION AND CONNECTION TO INFONCA

Unlike in the method section in the main text where we use the difference-based relationship to construct the loss, we can also construct the loss using ratio-based relationship.

Eq. (1) implies that, up to the (unknown) partition function $Z(c)$, the exponentiated reward induces an energy-based density over responses proportional to a power of the optimal density ratio. Concretely, rewriting in terms of a probability ratio yields

$$\frac{\exp(r(c,x))}{Z(c)^\beta} \;=\; \left(\frac{p_{\theta^*}(x \mid c)}{p_{\mathrm{ref}}(x \mid c)}\right)^\beta. \tag{28}$$

This suggests viewing both the true reward r and the model-induced reward r_θ as defining *energies* over a finite candidate set. Given a context c and a set of candidates $\{x_i\}_{i=1}^K$, introduce a generic nonnegative weighting scheme $\{w_i\}_{i=1}^K$ (e.g., importance weights, sampling correction, or uniform weights) and define the corresponding normalized distributions

$$\begin{aligned} q^*(i \mid c, \{x\}) \;&\triangleq\; \frac{w_i \,\exp\big(r(c,x_i)/\beta\big)}{\sum_{j=1}^K w_j \,\exp\big(r(c,x_j)/\beta\big)}, \\ q_\theta(i \mid c, \{x\}) \;&\triangleq\; \frac{w_i \,\exp\big(r_\theta(c,x_i)/\beta\big)}{\sum_{j=1}^K w_j \,\exp\big(r_\theta(c,x_j)/\beta\big)}. \end{aligned} \tag{29}$$

When $r_\theta = r$ (up to an additive constant independent of x), these two distributions coincide:

$$q^*(i \mid c, \{x\}) = q_\theta(i \mid c, \{x\}). \tag{30}$$

Crucially, the normalization in Eq. (29) removes dependence on the unknown $Z(c)$, converting reward matching into distribution matching over a finite set.

Divergence minimization view. A natural objective is therefore to fit q_θ to q^* by minimizing a divergence over sets:

$$\mathcal{L}_{\mathrm{dist}}(\theta) \;=\; \mathbb{E}_{\rho(c,\{x_i\})}\Big[D\big(q^*(\cdot \mid c, \{x\}) \,\|\, q_\theta(\cdot \mid c, \{x\})\big)\Big], \tag{31}$$

where $D(\cdot\|\cdot)$ can be chosen as forward KL, reverse KL, Jensen–Shannon, or other proper scoring rules.

Special case: uniform weights and forward KL recover InfoNCA. For the common choice $w_i \equiv 1$, both q^* and q_θ are categorical distributions on $\{1, \ldots, K\}$ (Oord et al., 2018). Taking D to be the forward KL gives

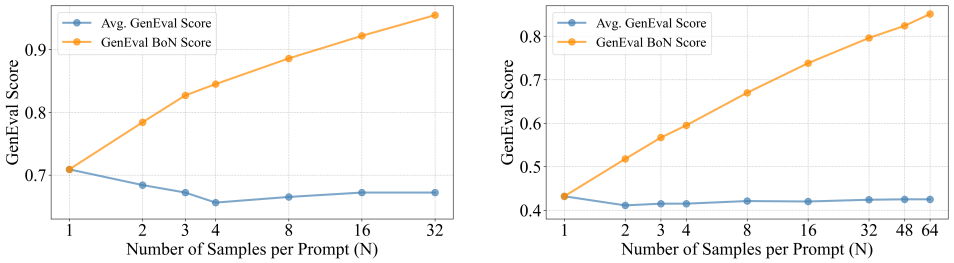
$$\begin{aligned} \mathcal{L}_{\mathrm{InfoNCA}}(\theta) &\triangleq \mathbb{E}_{\rho(c,\{x_i\})}\Big[\mathrm{KL}\big(q^*(\cdot \mid c, \{x\}) \,\|\, q_\theta(\cdot \mid c, \{x\})\big)\Big] \\ &= \mathbb{E}_{\rho(c,\{x_i\})}\Big[-\sum_{i=1}^K q^*(i \mid c, \{x\}) \log q_\theta(i \mid c, \{x\})\Big] \;+\; \text{const.} \end{aligned} \tag{32}$$

Thus, minimizing forward KL is equivalent to minimizing cross-entropy from the "teacher" distribution induced by r to the "student" distribution induced by r_θ.

H ADDITIONAL RESULTS

H.1 BEST-OF-N (BON) PERFORMANCE OF SD3.5-M AND SD1.5

Best-of-N (BoN) is an inference-time scaling strategy that generates N independent samples and selects the one with the highest reward score (Gao et al., 2023; Beirami et al., 2024; Eisenstein et al., 2023; Gui et al., 2024; Ma et al., 2025a). We report the BoN performance of SD3.5-M and SD1.5 on `GenEval` in Fig. 6 as a reference. The orange curve reports BoN performance as N increases (log-linear for both SD3.5-M and SD1.5), while the blue curve shows the average reward score on the current N samples. The results suggest that RL fine-tuning can effectively amortize this inference-time cost, matching or exceeding BoN performance without requiring multiple samples at test-time.



(a) BoN results with SD3.5-M (b) BoN results with SD1.5

Figure 6: BoN results. The orange curve is the BoN performance, and the blue curve is the average performance of the current N samples.

Table 4: **Performance on Human Preference benchmarks with Multi-reward Training.** Task metrics, image quality and preference scores are all evaluated on `DrawBench` (Saharia et al., 2022) prompts. Baseline results are taken from Flow-GRPO (Liu et al., 2025b) and Choi et al. (2026), or reimplemented when unavailable in the original papers. Flow-GRPO is trained on `PickScore` only. `ImgRwd`: `ImageReward`.

Model	Task Metric			Preference Score	
	`PickScore` ↑	`CLIPScore` ↑	`HPSv2.1` ↑	`Aesthetics` ↑	`ImgRwd` ↑
SD3.5-M	22.34	27.99	0.279	5.39	0.87
Flow-GRPO (Liu et al., 2025b) (w/o KL)	23.41	—	—	6.15	1.24
Flow-GRPO (Liu et al., 2025b) (w/ KL)	23.31	27.81	0.315	5.92	1.28
DiffusionNFT (Zheng et al., 2025)	23.61	28.80	0.344	6.04	1.46
Choi et al. (2026)	23.68	29.60	0.325	6.06	1.45
CRD	23.27	29.63	0.321	5.70	1.35
+ CFG sampling=1.5	23.22	29.91	0.323	5.60	1.39
+ CFG sampling=3.0	22.89	30.01	0.315	5.47	1.36

H.2 Multi-reward Training

We also evaluate our method under a multi-reward training setting, where the model is jointly optimized using multiple rewards. Following prior work (Zheng et al., 2025; Choi et al., 2026), we report `PickScore`, `CLIPScore`, and `HPSv2.1` as task metrics and trained on the `PickScore` prompt dataset, alongside `Aesthetics` and `ImageReward` (`ImgRwd`) as preference scores, all evaluated on `DrawBench` (Saharia et al., 2022) prompts. As shown in Tab. 4, CRD achieves competitive performance across all metrics compared to baselines such as Flow-GRPO (Liu et al., 2025b) and Choi *et al.* (Choi et al., 2026). Notably, applying CFG sampling further improves `CLIPScore`, suggesting better text alignment, although at a slight cost to aesthetic quality at higher CFG values. Corresponding visualizations can be found in Fig. 14.

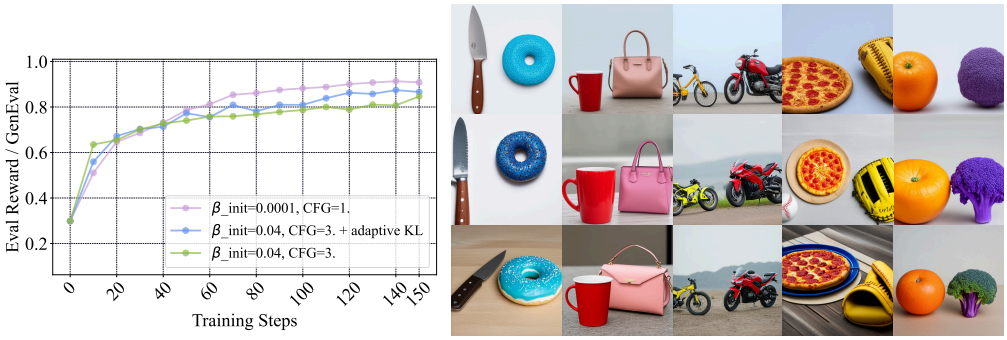
H.3 Adaptive KL Regularization on DiffusionNFT

In Fig. 7, we demonstrate that the proposed Adaptive KL Regularization also effectively mitigates reward hacking when applied to DiffusionNFT, preserving visual quality (fewer white backgrounds) while maintaining high reward scores.

H.4 Additional Ablations

Temperature in CRD objective. As shown in Fig. 8, varying τ over a wide range (1.0, 10.0, 100.0, ∞) yields remarkably similar reward curves across both small and large β_{init}, suggesting that CRD is largely insensitive to the choice of τ. For simplicity, we use uniform weighting (*i.e.* $\tau{=}\infty$) in all other experiments.

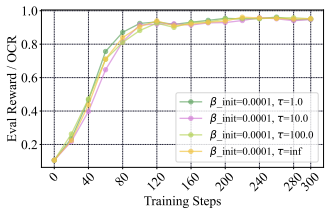
Group size K. Fig. 9 shows a clear trend: larger group sizes lead to faster convergence and higher final rewards, with $K{=}24$ achieving near-perfect OCR scores by the end of training, as larger groups provide richer gradient signal per update. Notably, even $K{=}2$ yields meaningful learning, demonstrating the robustness of CRD under minimal sampling budgets.



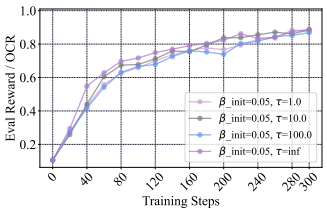
(a) Evaluation `GenEval` score during training, under different β_{init} and CFG values.

(b) Visual comparison, from top to bottom corresponding to the pink, blue and green curve in Fig. 7a.

Figure 7: Effectiveness of adaptive initial KL on DiffusionNFT.



(a) β_{init}=0.0001; CFG=1.0



(b) β_{init}=0.05; CFG=3.0

Figure 8: Ablations on different temperatures τ

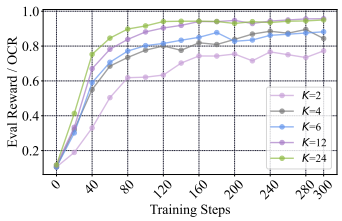
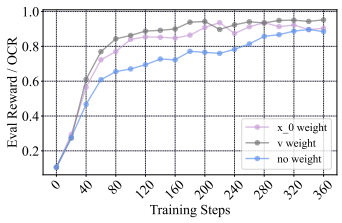
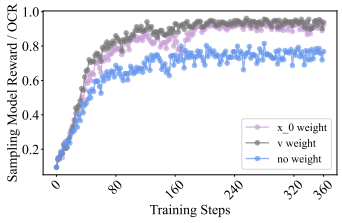


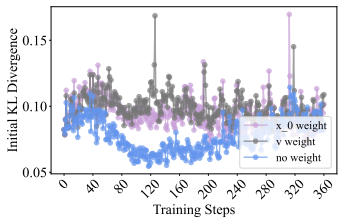
Figure 9: Ablation on group size K with OCR reward



(a) Evaluation reward of the current model θ



(b) Reward value during training of the sampling model p_{samp}



(c) Initial KL value during training

Figure 10: Ablations on adaptive weighting in ELBO estimator

Adaptive weighting in ELBO estimator. The original adaptive weighting (Yin et al., 2024; Zheng et al., 2025) is built on the x_0 representation, which introduces an extra coefficient t compared to the estimator in Eq. (13):

$$\widehat{R}_\theta^{\text{adap_}x_0}(c,x) \triangleq -\beta\,\mathbb{E}_{t,\epsilon}\left[\frac{td\,\|v_\theta(x_t,t\mid c)-v_{\text{target}}\|_2^2}{\text{sg}(\|v_\theta(x_t,t\mid c)-v_{\text{target}}\|_1)}-\frac{td\,\|v_{\text{ref}}(x_t,t\mid c)-v_{\text{target}}\|_2^2}{\text{sg}(\|v_{\text{ref}}(x_t,t\mid c)-v_{\text{target}}\|_1)}\right]. \tag{33}$$

As shown in Fig. 10, our v-prediction adaptive weighting (Eq. (13)) achieves the highest evaluation reward and sampling model reward, while maintaining stable KL divergence throughout training. We therefore adopt our adaptive weighting scheme in all experiments.

H.5 Additional Qualitative Results

Visualization during training process We visualize generated samples at successive checkpoints throughout training, starting from step 0 and sampled at every 200-step interval. As training pro-

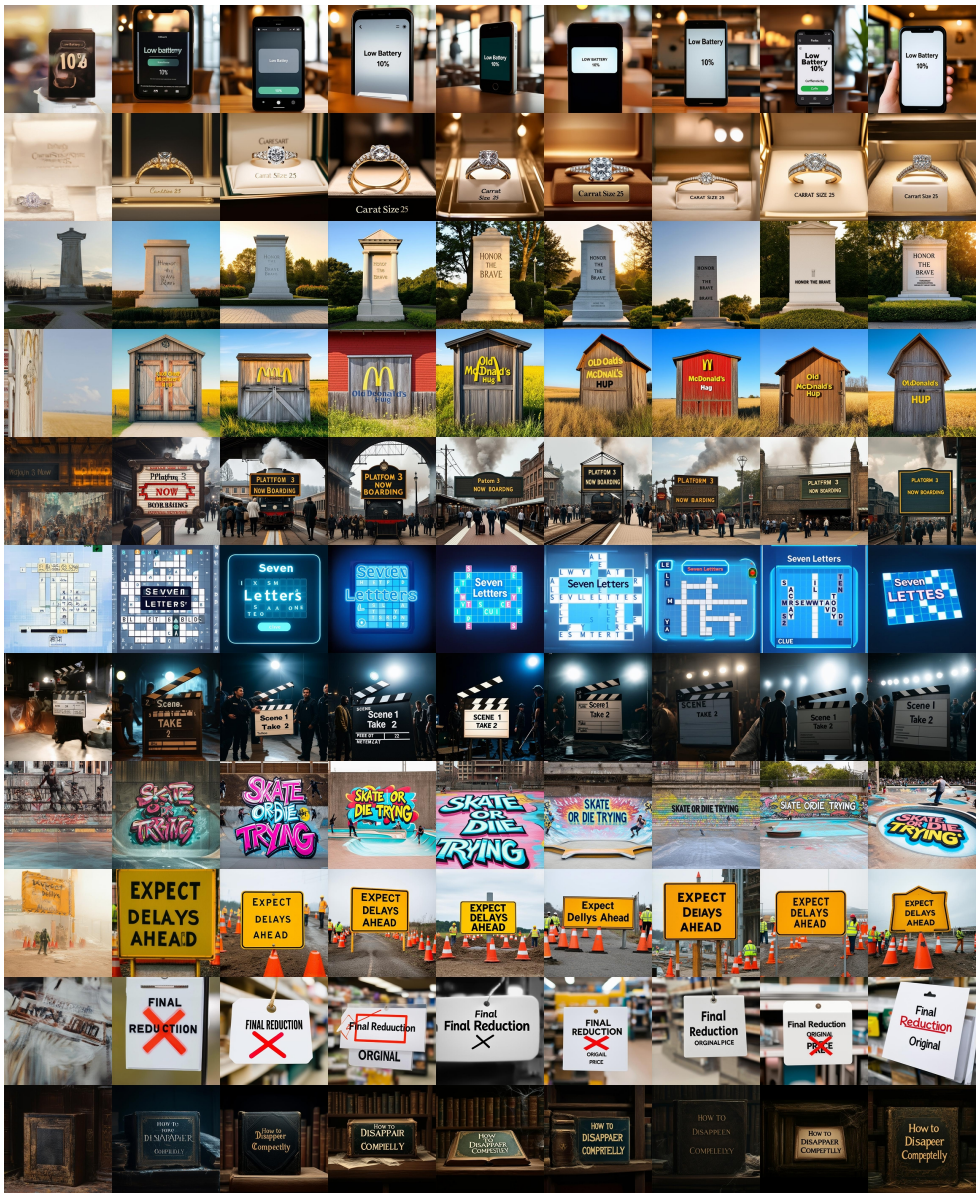


Figure 11: We visualize samples generated with our method at successive checkpoints (from left to right) throughout training, starting from step 0 and sampled at every 200-step interval.

gresses, the model learns to render text more accurately within the generated images without collapsing, reflecting steady improvements in text fidelity over time.

Inference with higher CFG In Fig. 12, we compare samples generated with increasing CFG scales. While moderate CFG typically improves prompt adherence, we observe a consistent degradation in visual fidelity as CFG becomes large. For text-centric prompts, higher CFG often introduces typographic artifacts such as broken glyphs, and unstable letter shapes, even when the overall image appears more strongly aligned with the prompt. For compositional prompts, increasing CFG does not reliably improve prompt following and can even reduce it. For example, the *four* stop sign prompt frequently violates the counting constraint at the higher CFG. Finally, we observe a style shift at large CFG values, where the cow in the last column gradually transitions from a photorealistic appearance to a more stylized, illustration-like rendering with simplified textures and exaggerated features.



Figure 12: Visualization of generation with different inference CFG values. The left 5 columns are generated with prompts from `GenEval` and the right 5 columns are generated with prompts from `OCR` testsets, respectively.



Figure 13: **Qualitative visual generations.** Each row shows generations from a different random seed for the same prompt. Prompts follow the template "a photo of a {} zebra", where the placeholder is replaced by the colors red, blue, green, yellow, and purple from top to bottom.

Colored zebra. We further demonstrate the model's ability to generate zebras in a variety of colors. The training dataset has a roughly uniform color distribution:

```
Counter({'white': 1857, 'purple': 1817, 'green': 1797, 'brown': 1782,
  'black': 1771, 'orange': 1749, 'pink': 1742, 'blue': 1722, 'red':
                        1715, 'yellow': 1696}).
```

Qualitative generations with simple prompts are shown in Fig. 13. We highlight three observations: (*1*) the model produces diverse and visually distinct outputs across colors; (*2*) despite the balanced color distribution in training, the model struggles to generate convincing blue and green zebras (failure generation); and (*3*) in most cases, the black stripes are replaced by the target color, with only rare cases where the zebra retains its black stripes alongside the assigned color.

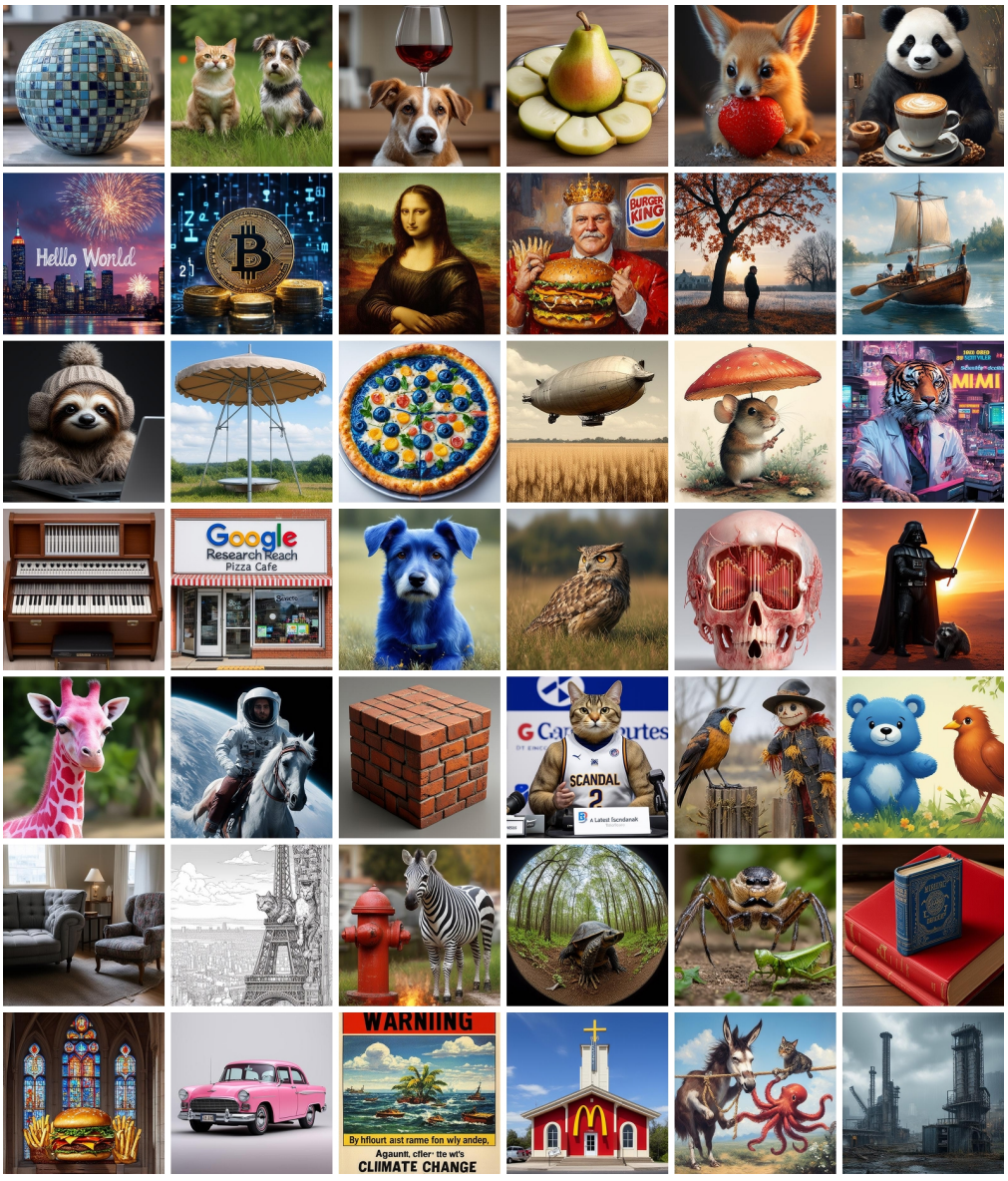


Figure 14: Qualitative visual generations from our model trained with multiple rewards. The text prompts are selected from the `DrawBench`.