FLOWAWR: ONLINE ADAPTIVE FLOW REINFORCEMENT VIA ADVANTAGE-WEIGHTED RECTIFICATION

Anonymous authors
Paper under double-blind review



Figure 1: Samples generated by FLUX.1-dev optimized with FlowAWR using HPSv3 as the reward. Prompts are drawn from the HPDv2 and PickScore test sets.

ABSTRACT

Aligning generative flow models on continuous spaces via online reinforcement learning is constrained by intractable trajectory likelihoods. Existing density-approximated policy gradient methods rely on stochastic SDE samplers to construct tractable transition kernels, which introduce training-inference inconsistencies and necessitates Classifier-Free Guidance (CFG). DiffusionNFT bypasses trajectory likelihoods through implicit velocity regression, but uses fixed residual coefficients in its branch targets, with rewards controlling their relative loss weights. We propose *Flow Advantage-Weighted Rectification* (**FlowAWR**), a paradigm that recasts continuous generative policy optimization as supervised regression toward a theoretically optimal velocity field. Starting from the optimal policy of a KL-constrained reward maximization, FlowAWR derives the optimal velocity field that admits a magnitude-aware, advantage-weighted rectification form, yielding SDE-free optimization. In comparative evaluations on SD3.5-Medium, FlowAWR achieves improved alignment performance alongside a $2\times$ to $5\times$ convergence acceleration over DiffusionNFT (e.g., reaching a 24.12 PickScore in 1.2k steps, versus 23.82 in 2.0k steps for DiffusionNFT and 23.50 in $>$4k steps for FlowGRPO).

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.

Under multi-reward constraints, FlowAWR sustains quality, satisfying structural rules with stable out-of-domain performance.

1 INTRODUCTION

The empirical success of Reinforcement Learning (RL) in the post-training of Large Language Models (LLMs) (Ziegler et al., 2019; Achiam et al., 2023; Comanici et al., 2025; Yang et al., 2025) has motivated its extension to continuous generative spaces. Within this domain, Flow Matching (Lipman et al., 2022; Liu et al., 2022) has emerged as a prevalent paradigm, offering a deterministic trajectory framework for modern diffusion models (Esser et al., 2024; Labs et al., 2025; Seedream et al., 2025; Wu et al., 2025). However, adapting RL to this continuous space introduces a structural constraint: the approximation of trajectory likelihood. Unlike AR models that compute exact discrete token probabilities, continuous flow models inherently lack tractable step-wise transition densities (Song et al., 2020). Ranging from Direct Preference Optimization (DPO) approaches (Wallace et al., 2024; Yang et al., 2024) to recent online policy gradient approaches (Liu et al., 2025; Xue et al., 2025b), this likelihood bottleneck consistently compromises the computational efficiency and theoretical guarantees of RL on continuous spaces.

To circumvent this likelihood barrier, a prevalent strategy discretizes the continuous generative process into a multi-step Markov Decision Process (MDP) (Black et al., 2023; Fan et al., 2023). Building upon this, recent GRPO-based methods (Liu et al., 2025; Xue et al., 2025b) inject stochasticity via SDE samplers to construct tractable Gaussian transition kernels, enabling the evaluation of policy gradients. Although subsequent improvements attempt to interleave ODE and SDE steps (Li et al., 2025), this paradigm fails to resolve the forward inconsistency between stochastic training trajectories and deterministic inference rollouts. Furthermore, by failing to implicitly distill the guidance signal into the vector field, these methods retain a structural reliance on CFG (Ho & Salimans, 2022) to maintain generation quality, thereby imposing computational overhead during inference.

An alternative paradigm leverages implicit optimization and velocity field guidance to bypass explicit density estimation. Early reward-weighted methods (Lee et al., 2023) are typically restricted to offline settings and lack explicit penalization mechanisms for low-quality samples. Recent online RL frameworks, such as DiffusionNFT (Zheng et al., 2025), have reformulated policy optimization as a supervised velocity regression task, achieving notable improvements in both training efficiency and generation performance over GRPO-style methods. As our analysis shows, its implicit positive and negative velocity fields induce two regression targets with fixed residual coefficients, while normalized rewards determine their relative loss weights. This leaves open how to derive sample-dependent target corrections directly from the reward-maximizing policy.

Herein, we propose Flow Advantage-Weighted Rectification (**FlowAWR**), an online RL framework that derives velocity regression targets from reward-maximizing policies without introducing stochastic sampling trajectories. Our framework originates from the exact closed-form optimal policy derived from reward maximization under a KL penalty relative to the reference policy. By mapping this optimal terminal distribution onto the intermediate probability paths, we analytically derive the corresponding optimal velocity field. We demonstrate that this optimal field is equivalent to rectifying the reference velocity field by an advantage-weighted expectation. Consequently, FlowAWR recasts flow RL as a supervised regression task toward this theoretical optimum. In practice, a group-based approximation to the derived advantage uses relative rewards to determine sample-dependent residual coefficients in the regression targets.

Our method preserves the architectural benefits of likelihood-free approaches by eliminating the forward inconsistency induced by SDE sampling, while its advantage-weighted updates implicitly distill alignment trajectories into the network parameters to enable CFG-free generation. To evaluate the efficacy of FlowAWR, we conduct systematic experiments on the SD3.5-Medium (2.5B parameters) model (Esser et al., 2024). Quantitative results indicate that, in the single-reward setting, FlowAWR achieves a $2\times$ to $5\times$ convergence acceleration alongside improved performance compared to DiffusionNFT. For instance, when optimizing for PickScore, FlowAWR elevates the metric to 24.12 within 1.2k steps. In contrast, DiffusionNFT requires 2.0k steps to reach 23.82, and FlowGRPO necessitates over 4k steps while relying on CFG to achieve 23.50. Furthermore, under the complex distribution shifts induced by multi-reward optimization, our sequential multi-reward training maintains optimization stability and mitigates out-of-domain aesthetic degradation.

Our primary contributions are summarized as follows:

- We introduce **FlowAWR**, an SDE-free online RL framework based on velocity regression, and provide a **theoretical interpretation** of DiffusionNFT's implicit positive and negative velocity fields through their shared residual-rectification structure.
- We characterize the **optimal velocity field** by propagating the closed-form optimal policy to intermediate optimal distributions along the probability path, ultimately instantiating this theoretical optimum via a **group-based approximated advantage function**.
- We conduct quantitative evaluations across both single- and multi-objective optimization settings, indicating that FlowAWR accelerates convergence rates and improves empirical alignment performance compared to baseline methodologies.

2 Background

2.1 Flow Matching and Rectified Flow

For noise $x_0 \sim p_0$ and data $x_1 \sim p_1$, we adopt the linear path of Flow Matching and Rectified Flow (Lipman et al., 2022; Liu et al., 2022):

$$x_t = tx_1 + (1-t)x_0, \qquad t \in [0,1].$$

Each pair defines a constant interpolation velocity, $u_t(x_t|x_1) = x_1 - x_0 = (x_1 - x_t)/(1-t)$ for $t < 1$. The model $v_\theta(x_t, t)$ is trained to regress this target:

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{t\sim\mathcal{U}[0,1],x_1\sim p_1,x_0\sim p_0}\left[\|v_\theta(x_t,t) - u_t(x_t|x_1)\|^2\right]. \tag{1}$$

At inference, the learned field generates samples by integrating the deterministic ODE $\mathrm{d}x_t = v_\theta(x_t,t)\,\mathrm{d}t$ from the noise prior. These model trajectories need not coincide with the linear interpolations used for training.

At the population optimum, velocity regression recovers the posterior average of the conditional fields (Lipman et al., 2022; Albergo et al., 2025). Writing the posterior data mean as $\hat{x}_1(x_t) = \mathbb{E}[x_1 \mid x_t]$, the marginal velocity is

$$v(x_t,t) = \mathbb{E}\left[u_t(x_t|x_1) \mid x_t\right] = \frac{\hat{x}_1(x_t) - x_t}{1-t}. \tag{2}$$

2.2 Reinforcement Learning as Supervised Fine-tuning

DiffusionNFT (Zheng et al., 2025) treats RL as supervised velocity regression rather than trajectory-level policy optimization. It uses reward feedback to distinguish positive and negative components of the reference policy, represented implicitly by mixing the trainable field v_θ with the reference field v^{old}. Omitting their shared arguments (x_t, t), the mixed velocities are

$$v_\theta^{\pm} := v^{\text{old}} \pm \beta(v_\theta - v^{\text{old}}), \tag{3}$$

where the fixed scale β controls their opposite deviations from the reference. For a terminal sample x_1 with normalized reward $r \in [0,1]$, the two fields are regressed toward $u_t = u_t(x_t|x_1)$ through

$$\mathcal{L}_{\text{NFT}}(\theta) = \mathbb{E}\left[r\|v_\theta^+ - u_t\|^2 + (1-r)\|v_\theta^- - u_t\|^2\right]. \tag{4}$$

Higher rewards emphasize positive-field regression, while lower rewards emphasize the negative-field term. Both terms update the same trainable field v_θ through the coupled construction above, incorporating reward feedback without evaluating trajectory-level likelihoods.

3 Method

3.1 Revisiting Optimal Policy from Fixed Experience Buffer

At each online iteration, samples from the current policy π^{old} form an experience buffer $\mathcal{D}$. We hold the buffer and reference policy fixed during policy optimization and refresh them between iterations.

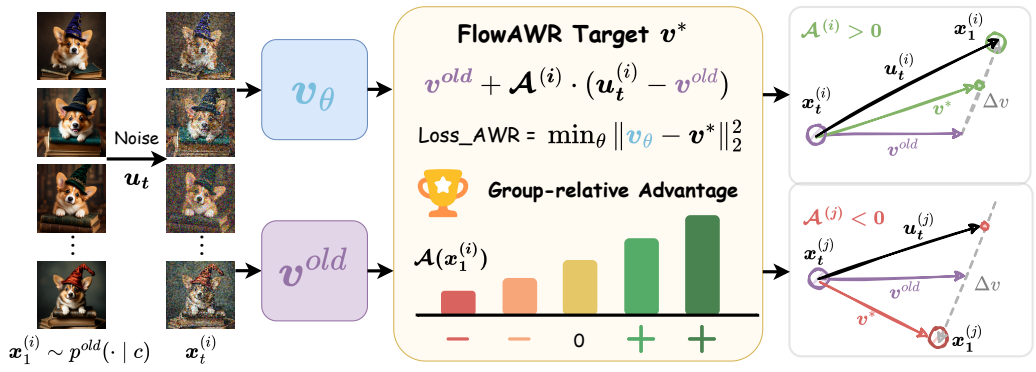


Figure 2: **Overview of FlowAWR.** The model regresses v_θ toward $v^{\text{old}} + \mathcal{A}(u_t - v^{\text{old}})$, using group-relative advantages to control the direction and magnitude of velocity rectification.

For this fixed-reference step, we consider KL-regularized reward maximization (Peters et al., 2010; Schulman et al., 2015; Levine, 2018; Peng et al., 2019):

$$\max_{\pi_\theta} \mathbb{E}_{s\sim\mathcal{D}} \left[\mathbb{E}_{a\sim\pi_\theta(\cdot|s)} R(s,a) - \gamma D_{\text{KL}}(\pi_\theta(\cdot|s) || \pi^{\text{old}}(\cdot|s)) \right], \tag{5}$$

Here s and a denote the state and action, $R(s,a)$ the reward, and $\gamma > 0$ the KL penalty weight.

Optimizing over normalized action distributions yields a closed-form policy (Peters et al., 2010; Rafailov et al., 2023). With $Z(s)$ denoting the state-dependent partition function, the solution is

$$\pi^*(a|s) = \frac{1}{Z(s)} \pi^{\text{old}}(a|s) \exp\left(\frac{R(s,a)}{\gamma}\right). \tag{6}$$

The derivation is provided in Appendix A.1. This solution reweights the reference policy toward higher-reward actions, with γ controlling the strength of the reward tilt. And for generative Flow Matching, the state s corresponds to the prompt c and the action a to the generated sample x_1. The optimal policy therefore specifies a target terminal distribution.

3.2 Derivation of Optimal Velocity Field

Building upon the optimal policy in Eq. equation 6, we first identify the optimal terminal distribution $p_1^*(x_1|c)$, which represents the theoretical optimum for reward maximization on the current experience buffer, regardless of the exploration mechanism (e.g., SDE-based policy optimization):

$$p^*(x_1|c) = \frac{1}{Z(c)} p^{\text{old}}(x_1|c) \exp\left(\frac{R(c,x_1)}{\gamma}\right),$$

where c denotes a prompt, and $p^{\text{old}}(x_1|c)$ represents the reference policy. To derive the corresponding velocity field, the explicit form of the intermediate marginal distribution $p^*(x_t|c)$ is required:

Proposition 1 (Intermediate Marginals). *Given the optimal terminal distribution p_1^*, the corresponding marginal density at t relates to the reference density p_t^{old} via a value function $\phi(x_t, c, t)$:*

$$p_t^*(x_t|c) = \frac{1}{Z(c)} p_t^{\text{old}}(x_t|c) \cdot \phi(x_t, c, t), \tag{7}$$

where $\phi(x_t, c, t) = \mathbb{E}_{x_1\sim p^{\text{old}}(\cdot|x_t,c)} \left[\exp\left(\frac{R(c,x_1)}{\gamma}\right)\right]$. Proof in Appendix A.2.

With the density relationship established, the explicit form of the optimal velocity field emerges. Combining the velocity form in Eq. equation 2 with Tweedie's formula (Efron, 2011), which links the posterior mean to the marginal score via $\hat{x}_1(x_t, c) = \frac{x_t}{t} + \frac{(1-t)^2}{t} \nabla_{x_t} \log p_t(x_t|c)$, we observe that the velocity is strictly determined by the score function of $p_t^*(x_t|c)$ in Eq. equation 7:

$$\begin{aligned} v^*(x_t, c, t) &= \frac{x_t}{t} + \frac{1-t}{t} \nabla_{x_t} \log p_t^*(x_t|c) \\ &= \frac{x_t}{t} + \frac{1-t}{t} \nabla_{x_t} \left[\log p_t^{\text{old}}(x_t|c) + \log \phi(x_t, c, t)\right] \\ &= v^{\text{old}}(x_t, c, t) + \frac{1-t}{t} \nabla_{x_t} \log \phi(x_t, c, t). \end{aligned} \tag{8}$$

While Eq. equation 8 provides a theoretically rigorous direction for rectification, directly computing $\nabla_{x_t} \log \phi(x_t, c, t)$ is intractable. Herein, we expand the gradient into an expectation over samples, which reveals a connection to advantage-weighted regression (Peng et al., 2019; Zhang et al., 2025).

Theorem 1 (**Advantage-Weighted Rectification**). *The optimal velocity field $v^*(x_t, c)$ derived in Eq. equation 8 can be equivalently expressed as the reference field rectified by an advantage-weighted residual expectation (proof in Appendix A.4):*

$$v^*(x_t, c, t) = v^{\text{old}}(x_t, c, t) + \mathbb{E}_{x_1 \sim p^{\text{old}}(\cdot|x_t, c)} \left[\mathcal{A}(x_1, x_t) \left(u_t(x_t|x_1) - v^{\text{old}}(x_t, c, t) \right) \right],$$

where $\mathcal{A}(x_1, x_t)$ is the Centered Advantage, constructed by leveraging the zero-expectation property $\mathbb{E}[u_t - v^{\text{old}}] = 0$ to subtract a constant baseline of 1:

$$\mathcal{A}(x_1, x_t) = \frac{\exp\left(\frac{R(c,x_1)}{\gamma}\right)}{\phi(x_t, c, t)} - 1. \tag{9}$$

This formulation reveals that the optimal field v^* is obtained by dynamically rectifying the reference field v^{old} with the velocity residual $(u_t - v^{\text{old}})$ based on $\mathcal{A}$. This exhibits an explicit awareness of sample quality, enabling the formulation to quantitatively differentiate between positive and negative samples through the polarity of $\mathcal{A}$, thereby reinforcing or repelling trajectories accordingly.

3.3 Flow Reinforcement via Advantage-Weighted Rectification

Based on Theorem 1, the policy optimization process can be cast as a supervised regression task towards the optimal velocity field. Ideally, the formulation is the minimization of the distance between v_θ and the theoretical optimum v^*, as defined in the standard objective in Eq. equation 1:

$$\mathcal{L}_{\text{Ideal}}(\theta) = \mathbb{E}_{t, x_t \sim p_t^{\text{old}}} \left[\left\| v_\theta(x_t, c, t) - v^*(x_t, c, t) \right\|^2 \right].$$

However, directly optimizing $\mathcal{L}_{\text{Ideal}}$ is intractable due to the expectation over $p^{\text{old}}(x_1|x_t, c)$ embedded within $v^*(x_t, c, t)$. To circumvent this, we propose the Advantage-Weighted Rectification (AWR) Loss, which learns from individual samples by constructing a stochastic rectified target:

$$\begin{aligned} \mathcal{L}_{\text{AWR}}(\theta) = \mathbb{E}_{t \sim \mathcal{U}[0,1], x_1 \sim p^{\text{old}}(\cdot|c), x_0 \sim p_0(\cdot)} \Bigg[\\ \left\| v_\theta(x_t, c, t) - \Big(v^{\text{old}}(x_t, c, t) + \mathcal{A}(x_1, x_t) \cdot \big(u_t(x_t|x_1) - v^{\text{old}}(x_t, c, t) \big) \Big) \right\|^2 \Bigg]. \end{aligned} \tag{10}$$

Theorem 2 (**Optimization Equivalence**). *The Advantage-Weighted Rectification Loss $\mathcal{L}_{\text{AWR}}(\theta)$ and the Ideal Loss $\mathcal{L}_{\text{Ideal}}(\theta)$ share the same global optimum. Specifically, the gradients of both objectives with respect to θ are identical:*

$$\nabla_\theta \mathcal{L}_{\text{AWR}}(\theta) = \nabla_\theta \mathcal{L}_{\text{Ideal}}(\theta).$$

which indicates that minimizing $\mathcal{L}_{\text{AWR}}$ drives the parameterized velocity field v_θ to converge to the theoretical optimal field $v^*(x_t, c, t)$. Proof in Appendix A.5.

Theorem 2 establishes a novel perspective for Flow RL: directly learning the optimal terminal distribution derived from each experience buffer. This results in a new off-policy paradigm that casts policy optimization as supervised regression on advantage-weighted samples (Figure 2).

In practice, we use **Group-Based Estimation** with G images $\{x_1^{(i)}\}_{i=1}^G$ sampled from the reference policy for each prompt c, following group-based methods (Liu et al., 2025; Zheng et al., 2025). We approximate the $\phi(x_t, c, t)$ by the within-prompt mean of exponentiated rewards, yielding:

$$\mathcal{A}(x_1^{(i)}) \approx \frac{\exp\left(\frac{R(c,x_1^{(i)})}{\gamma}\right)}{\frac{1}{G}\sum_{j=1}^{G} \exp\left(\frac{R(c,x_1^{(j)})}{\gamma}\right)} - 1. \tag{11}$$

Table 1: Our evaluation results compared with DiffusionNFT. Gray-colored : In-domain target task. The DiffusionNFT OCR row uses its official task-specific configuration ($\beta = 0.1$, delayed old-policy EMA); Appendix C.5 compares it with the general configuration.

Method	#Iter	Task Metric			DrawBench Metric				
		GenEval	OCR	PickScore	ClipScore	HPSv2.1	Aesthetic	ImgRwd	PickScore
SD3.5-M + CFG	—	0.63	0.59	21.75	0.285	0.279	5.36	0.85	22.34
DiffusionNFT	1k	**0.98**	0.35	20.19	0.274	0.240	4.99	0.40	21.78
FlowAWR	**0.5k**	**0.98**	0.37	20.25	0.274	0.252	5.09	0.43	21.92
DiffusionNFT	0.4k	0.20	**0.98**	19.15	0.233	0.168	4.79	-1.27	20.15
FlowAWR	**0.26k**	0.29	0.975	19.43	0.246	0.193	5.00	-0.78	20.51
DiffusionNFT	2.0k	0.55	0.67	23.82	0.272	0.316	6.19	1.31	24.09
FlowAWR	**1.2k**	0.61	0.72	**24.12**	0.275	0.315	6.28	1.36	24.29

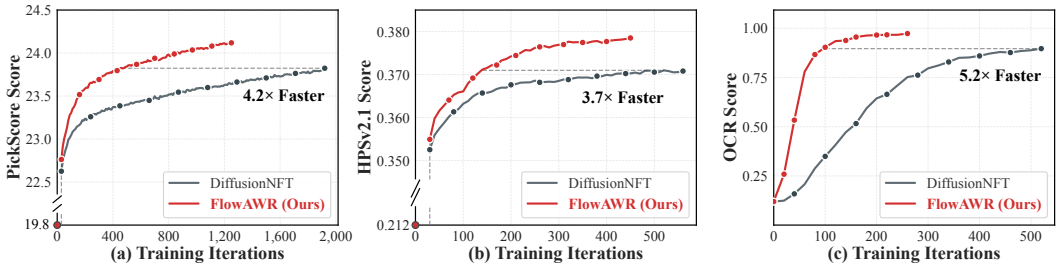


Figure 3: Head-to-head comparison between FlowAWR and DiffusionNFT on single-reward optimization tasks: (a) PickScore, (b) HPSv2.1, and (c) OCR.

Substituting these advantages into the AWR loss gives the group-averaged objective, with the shared conditioning c omitted from the velocity fields:

$$\mathcal{L}^{(c)}_{\mathrm{AWR}}(\theta) = \frac{1}{G}\sum_{i=1}^{G} \mathbb{E}_{t,x_0^{(i)}\sim p_0(\cdot),x_1^{(i)}\sim p_1^{\mathrm{old}}(\cdot|c)} \Bigg[\left\| v_\theta(x_t^{(i)},t) - \left(v^{\mathrm{old}}(x_t^{(i)},t) + \mathcal{A}(x_1^{(i)}) \cdot \left(u_t(x_t^{(i)}|x_1^{(i)}) - v^{\mathrm{old}}(x_t^{(i)},t) \right) \right) \right\|^2 \Bigg].$$

We apply **Adaptive Scaling** by setting γ to the standard deviation of rewards across the current experience buffer, adapting the KL penalty to the reward scale. For **Soft Online Evolution**, the reference parameters follow the EMA update $\theta^{\mathrm{old}} \leftarrow \mu_i\theta^{\mathrm{old}} + (1-\mu_i)\theta$, where μ_i is the decay rate at iteration i; its schedule is examined in Section 4.4. In our SD3.5-Medium experiments, we also use **CFG-Free Generation** during rollouts and inference, as in DiffusionNFT (Zheng et al., 2025).

3.4 Unified Perspective on Flow Reinforcement

Our FlowAWR framework not only provides a foundation for supervised flow policy optimization but also establishes a unifying lens through which both standard supervised fine-tuning and recent heuristic methods can be understood. Together, Theorems 1 and 2 yield the following sample-wise regression target for learning the optimal velocity field:

$$v^*(x_t,c,t) = v^{\mathrm{old}}(x_t,c,t) + \mathcal{A}(x_1,x_t)\cdot\left(u_t(x_t|x_1) - v^{\mathrm{old}}(x_t,c,t)\right). \tag{12}$$

Theoretical Interpretation of DiffusionNFT. DiffusionNFT's implicit positive and negative velocity fields induce regression targets that share the residual structure above. For a fixed $\beta > 0$, factoring out β from its coupled regression loss in Eqs. equation 3 and equation 4 gives

$$\mathcal{L}_{\mathrm{NFT}} \propto \mathbb{E}\left[r \left\| v_\theta - \left(v^{\mathrm{old}} + \frac{1}{\beta}(u_t - v^{\mathrm{old}}) \right) \right\|^2 + (1-r) \left\| v_\theta - \left(v^{\mathrm{old}} - \frac{1}{\beta}(u_t - v^{\mathrm{old}}) \right) \right\|^2 \right].$$

This factorization provides a theoretical interpretation of DiffusionNFT's implicit positive and negative velocity fields: their regression targets take the advantage-weighted rectification form in Eq. equation 12, with fixed coefficients $\mathcal{A}_{\mathrm{NFT}} = \pm 1/\beta$.

Table 2: Evaluations for the multi-reward setting. Gray-colored : In-domain reward. ‡Evaluated under 1024×1024 resolution. **Bold**: best result. Underline: second best.

Model	#Iter	Task Metric			DrawBench Metric				
		GenEval	OCR	PickScore	ClipScore	HPSv2.1	Aesthetic	ImgRwd	PickScore
SD-XL‡	—	0.55	0.14	21.74	0.287	0.280	5.60	0.76	22.42
SD3.5-L‡	—	0.71	0.68	22.24	0.289	0.288	5.50	0.96	22.91
FLUX.1-dev	—	0.66	0.59	22.55	0.295	0.274	5.71	0.96	22.84
SD3.5-M (w/o CFG)	—	0.24	0.12	19.90	0.237	0.204	5.13	-0.58	20.51
+ CFG	—	0.63	0.59	21.75	0.285	0.279	5.36	0.85	22.34
+ FlowGRPO	>5k	**0.95**	0.66	21.67	0.293	0.274	5.32	1.06	22.37
	2k	0.66	0.92	21.80	0.290	0.280	5.32	0.95	22.44
	4k	0.54	0.68	23.33	0.280	0.316	5.90	1.29	23.50
+ DiffusionNFT	1.7k	0.93	0.90	23.36	0.293	0.330	5.99	1.48	23.79
+ FlowAWR	1.5k	0.93	0.91	**23.60**	**0.296**	**0.341**	**6.04**	**1.51**	**23.91**

FlowAWR instead derives its rectification coefficient from the reward-tilted optimal policy. The theoretical coefficient normalizes the exponentiated reward by the conditional value function ϕ; in practice, we approximate this normalization using the within-prompt mean of exponentiated rewards. Combined with reward-adaptive γ, this construction determines sample-dependent rectification magnitudes without the separate mixing parameter β. The ablations in Section 4.4 support reward-adaptive scaling and show reward degradation in the fixed-coefficient single-branch variants.

Reduction to Supervised Fine-Tuning. Suppose all samples drawn from $p_1^{\text{old}}(\cdot \mid c)$ are treated as equally high-quality supervision and assigned the same reward. The normalized exponential reward then equals one. Without subtracting the unit baseline, the regression objective reduces to

$$\mathcal{L}(\theta) = \mathbb{E}\left[\left\|v_\theta - \left(v^{\text{old}} + 1 \cdot (u_t - v^{\text{old}})\right)\right\|^2\right] = \mathbb{E}\left[\|v_\theta - u_t\|^2\right].$$

4 Experiments

4.1 Experimental Setup

Our main experiments are based on SD3.5-Medium (Esser et al., 2024) at 512×512 resolution, with most experimental settings following FlowGRPO (Liu et al., 2025) and DiffusionNFT (Zheng et al., 2025). Additional validation on FLUX.1-dev (Labs et al., 2025) is provided in Appendix C.2.

Tasks and Evaluation Metrics. We categorize the text-to-image optimization tasks into two paradigms: (1) *Rule-based constraints*, utilizing GenEval (Ghosh et al., 2023) for compositional spatial reasoning and OCR accuracy for explicit text rendering. (2) *Model-based human preference*, employing PickScore (Kirstain et al., 2023) as the primary reward signal. To assess out-of-domain generalization, models are additionally evaluated on the DrawBench (Saharia et al., 2022) benchmark using an array of evaluation metrics: PickScore, CLIPScore (Hessel et al., 2021), HPSv2.1 (Wu et al., 2023), Aesthetic Score (Schuhmann, 2022), and ImageReward (Xu et al., 2023).

Training and Evaluation. We finetune with LoRA ($\alpha = 64, r = 32$). During training, each epoch processes 48 prompts, with an intra-group sample size of $G = 24$. For trajectory rollouts, we use $T = 10$ sampling steps for both head-to-head baseline comparisons, ablation studies and multi-reward joint training. All final evaluations are performed using a 40-step first-order ODE solver. Except in the EMA ablation, FlowAWR uses the same reference-policy EMA schedule, $\mu_i = \min(0.001i, 0.5)$, across all tasks. Further details and settings are provided in Appendix C.

4.2 Head-to-Head Comparison on Single Rewards

Table 1 and Figure 3 compare FlowAWR with DiffusionNFT on single-reward optimization. FlowAWR achieves comparable or higher target rewards with fewer updates on GenEval and PickScore. For OCR, the table uses DiffusionNFT's task-specific configuration, while the learning curves use its general configuration; Appendix C.5 provides the comparison. GPU-hour comparisons, including AWM and TempFlow-GRPO, are reported in Appendix C.1.

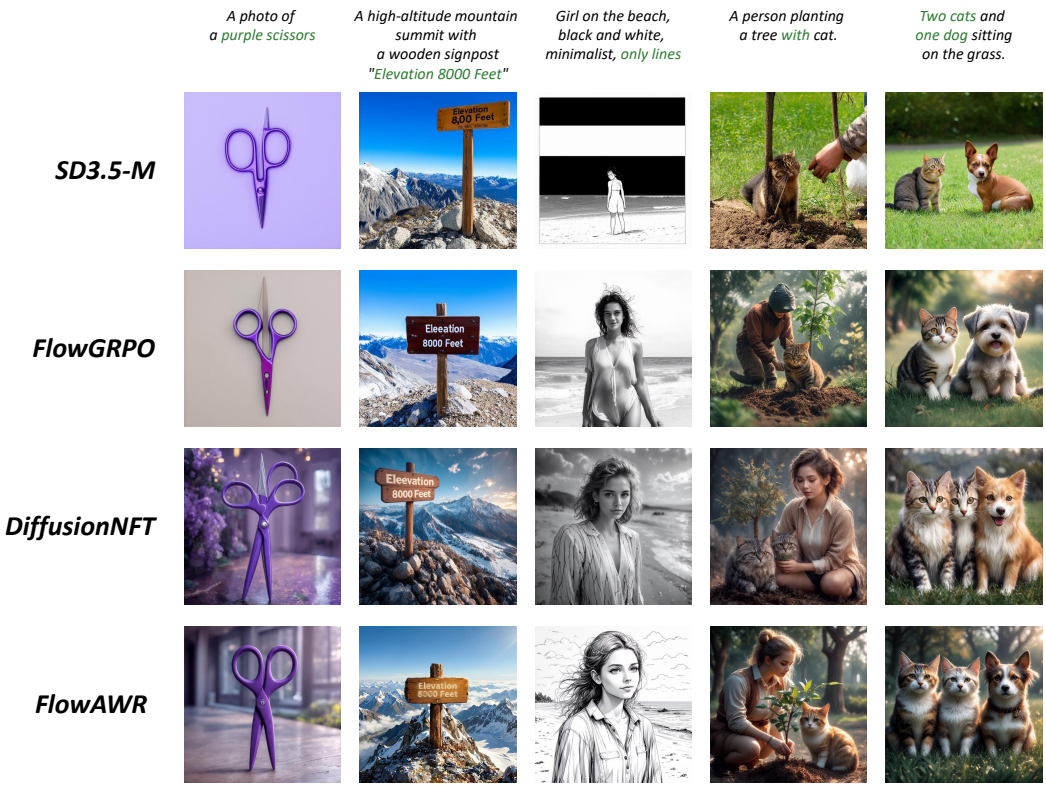


Figure 4: Qualitative Comparison. The text prompts are sampled from GenEval, OCR, PickScore, and DrawBench evaluations, illustrating the generation fidelity of our FlowAWR.

As evidenced in Table 1, FlowAWR exhibits reduced reward hacking during out-of-domain evaluations. However, exclusively optimizing a single rule-based objective (e.g., GenEval or OCR) typically degrades overall image aesthetics due to the absence of holistic visual constraints. To address this, we explore multi-reward sequential training in the subsequent section.

4.3 Multi-Reward Sequential Training

To mitigate the aesthetic degradation associated with isolated rule-based objectives, we extend FlowAWR to a multi-reward setting via a sequential optimization strategy. Using the CFG-free SD3.5-Medium, we first optimize the policy using a composite reward signal comprising PickScore, CLIPScore, and HPSv2.1 on the Pick-a-Pic dataset (Kirstain et al., 2023). We then continue training through stages that combine these preference rewards with GenEval or OCR, resuming the full training state at each transition (Appendix C.4). This sequential approach improves adherence to structural rules while preserving visual fidelity, as shown in Table 2 and Figure 4.

4.4 Ablation Study

Group Size. Figure 5(a,b) compares $G \in \{8, 24, 32\}$ on OCR and PickScore, with larger groups achieving higher final rewards. We adopt $G = 24$ to balance reward and sampling cost. **Adaptive γ Scaling.** The reward-adaptive choice achieves a higher final OCR reward than the fixed values $\gamma \in \{0.001, 0.01\}$ (Figure 5(c)), supporting adaptation to the reward scale. **Adaptive Advantage.** We compare adaptive advantages with fixed coefficients $\mathcal{A} \in \{\pm 1, \pm 10\}$, using one NFT-style regression target per sample and reward-dependent weights outside the squared loss. Both fixed settings suffer reward degradation on OCR and PickScore, whereas adaptive advantages yield increasing rewards (Figure 5(d,e)). **Soft Online Update.** Our default schedule $\mu_i = \min(0.001i, 0.5)$ achieves a higher OCR reward than the faster schedule $\mu_i = \min(0.01i, 0.9)$ (Figure 5(f)). On-policy updates ($\mu_i = 0$) improve reward initially but collapses.

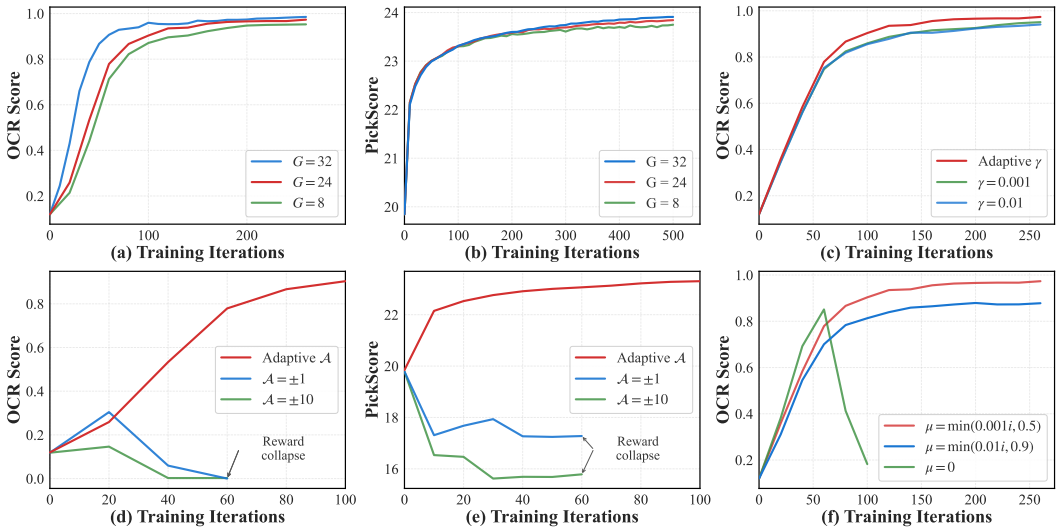


Figure 5: Ablation studies of FlowAWR. (a,b) Group size G on OCR and PickScore, respectively. (c) Adaptive versus fixed γ on OCR. (d,e) Adaptive versus fixed-magnitude advantages on OCR and PickScore, respectively. (f) Reference-policy update schedules μ_i on OCR.

5 Related Work

Implicit Optimization and Guidance-based Methods. Reward-based adaptation includes weighted regression, differentiable reward optimization, and inference-time guidance. Early RWR-style methods (Lee et al., 2023) favor high-reward samples through weighted or filtered fine-tuning. Differentiating rewards through sampling (Xu et al., 2023; Prabhudesai et al., 2023; Clark et al., 2023) requires differentiable feedback and incurs backpropagation costs, while inference-time guidance (Janner et al., 2022; Jin et al., 2025) adds computation during sampling without additional parameter updates at inference time. Recent methods connect reward optimization to pretraining objectives. Advantage Weighted Matching (AWM) (Xue et al., 2025a) formulates policy-gradient optimization through advantage-weighted score or flow matching. DiffusionNFT (Zheng et al., 2025) performs likelihood-free optimization through reward-weighted positive and negative velocity regression.

Density-Approximated Policy Gradient Methods. Trajectory-based methods formulate sampling as an MDP to apply standard RL algorithms (Black et al., 2023; Fan et al., 2023). Diffusion-DPO (Wallace et al., 2024) and D3PO (Yang et al., 2024) use likelihood approximations for preference optimization, while online methods such as DDPO (Black et al., 2023), FlowGRPO (Liu et al., 2025), and DanceGRPO (Xue et al., 2025b) optimize policies through stochastic transition likelihoods. For flow models, SDE sampling provides tractable transition densities but introduces differences between training trajectories and deterministic ODE inference. MixGRPO (Li et al., 2025) interleaves ODE and SDE steps to reduce training cost. TempFlow-GRPO (He et al., 2026) further branches trajectories at selected timesteps for localized credit assignment and applies noise-aware weighting to correct timestep-dependent gradient scaling. These refinements improve trajectory-based optimization while retaining stochastic transitions during training.

6 Conclusion

We introduce *Flow Advantage-Weighted Rectification* (**FlowAWR**), an off-policy online RL framework for continuous generative models. Starting from a KL-constrained reward maximization problem, FlowAWR derives an advantage-weighted velocity regression objective that recovers supervised fine-tuning under equal rewards without baseline subtraction. This analysis also provides a theoretical interpretation of DiffusionNFT's implicit positive and negative velocity fields, while motivating reward-derived, sample-dependent corrections to the regression targets. Empirically, FlowAWR achieves $2\times$ to $5\times$ faster convergence than DiffusionNFT in single-reward tasks, while remaining robust under the distribution shifts induced by multi-reward alignment.

AI USE STATEMENT

We used generative AI tools to assist with manuscript preparation, primarily for grammar correction, refinement of wording, and LaTeX formatting. AI-assisted checks do not replace author verification. The authors take responsibility for the final manuscript, including its technical claims and derivations.

ETHICS STATEMENT

Our experiments use publicly available pretrained models and established reward functions to evaluate FlowAWR. As a method for reward-based optimization of text-to-image models, FlowAWR may inherit biases from the underlying models and reward functions, and improvements in reward scores do not by themselves ensure safe or unbiased outputs. We have not identified additional ethical concerns specific to this study beyond these general risks.

REPRODUCIBILITY STATEMENT

We provide the derivation of the optimal policy in Appendix A.1 and proofs of the main theoretical results in Appendices A.2, A.4, and A.5. The experimental settings, advantage scaling, single-branch ablation configurations, and sequential multi-reward training procedure are described in Appendix C.4. Additional baseline comparisons and model-specific settings are reported in Appendix C, including the DiffusionNFT OCR configurations in Appendix C.5.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Michael Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025.

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

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

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011.

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.

Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36:79858–79885, 2023.

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.

Xiaoxuan He, Siming Fu, Yuke Zhao, Wanli Li, Jian Yang, Dacheng Yin, Fengyun Rao, and Bo Zhang. Tempflow-grpo: When timing matters for grpo in flow models. In *International Conference on Learning Representations*, volume 2026, pp. 132869–132895, 2026.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 7514–7528, 2021.

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

Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022.

Luozhijie Jin, Zijie Qiu, Jie Liu, Zijie Diao, Lifeng Qiao, Ning Ding, Alex Lamb, and Xipeng Qiu. Inference-time alignment control for diffusion models with reinforcement learning guidance. *arXiv preprint arXiv:2508.21016*, 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.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025.

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.

Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

Junzhe Li, Yutao Cui, Tao Huang, Yinping Ma, Chun Fan, Miles Yang, and Zhao Zhong. Mixgrpo: Unlocking flow-based grpo efficiency with mixed ode-sde. *arXiv preprint arXiv:2507.21802*, 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.

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*, 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.

Yuhang Ma, Yunhao Shui, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. Hpsv3: Towards wide-spectrum human preference score. *arXiv preprint arXiv:2508.03789*, 2025.

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, Katharina Mulling, and Yasemin Altun. Relative entropy policy search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, pp. 1607–1612, 2010.

Mihir Prabhudesai, Anirudh Goyal, Deepak Pathak, and Katerina Fragkiadaki. Aligning text-to-image diffusion models with reward backpropagation. *arXiv preprint arXiv:2310.03739*, 2023.

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.

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, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR, 2015.

Team Seedream, Yunpeng Chen, Yu Gao, Lixue Gong, Meng Guo, Qiushan Guo, Zhiyao Guo, Xiaoxia Hou, Weilin Huang, Yixuan Huang, et al. Seedream 4.0: Toward next-generation multimodal image generation. *arXiv preprint arXiv:2509.20427*, 2025.

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

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.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

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

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023.

Yixian Xu, Kaiyuan Gao, Yuxiang Chen, Yilei Chen, Zecheng Tang, Zihao Liu, Zikai Zhou, Deqing Li, Hao Meng, Kuan Cao, et al. Qwen-image-2.0-rl technical report. *arXiv preprint arXiv:2606.27608*, 2026.

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.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Kai Yang, Jian Tao, Jiafei Lyu, Chunjiang Ge, Jiaxin Chen, Weihan Shen, Xiaolong Zhu, and Xiu Li. Using human feedback to fine-tune diffusion models without any reward model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8941–8951, 2024.

Shiyuan Zhang, Weitong Zhang, and Quanquan Gu. Energy-weighted flow matching for offline reinforcement learning. *arXiv preprint arXiv:2503.04975*, 2025.

Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su, Stefano Ermon, Jun Zhu, and Ming-Yu Liu. Diffusionnft: Online diffusion reinforcement with forward process. *arXiv preprint arXiv:2509.16117*, 2025.

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 CONTENTS

A Extended Theoretical Derivations and Proofs **15**

A.1 Derivation of the Optimal Policy 15

A.2 Proof of Proposition 1 15

A.3 Derivation of Tweedie's Formula in Flow Matching 16

A.4 Proof of Theorem 1 17

A.5 Proof of Theorem 2 18

B Theoretical Discussions with RWR and DPO **19**

B.1 Connection to RWR in Flow Matching 19

B.2 Connection to DPO 20

C Experiment Details **20**

C.1 Training Cost and Additional Baseline Comparisons 20

C.2 Experiments on FLUX.1-dev 21

C.3 Empirical Range of the Centered Advantage 21

C.4 Experimental Setup 22

C.5 DiffusionNFT OCR Configurations 23

D Additional Qualitative Results **23**

A Extended Theoretical Derivations and Proofs

A.1 Derivation of the Optimal Policy

This section provides the analytical derivation of the closed-form optimal policy π^*, which is strictly optimized over the current fixed experience buffer $\mathcal{D}$ subject to a KL divergence constraint against the reference policy.

While the iterative update of π^{old} enables online RL, the step-wise optimization of π_θ fundamentally solves a constrained reward maximization problem over the empirical states in $\mathcal{D}$:

$$J(\pi_\theta) = \mathbb{E}_{s\sim\mathcal{D}}\left[\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}[R(s,a)] - \gamma D_{\text{KL}}(\pi_\theta(\cdot|s)||\pi^{\text{old}}(\cdot|s))\right],$$

where maximizing the global objective over $\mathcal{D}$ is mathematically equivalent to solving the optimization for each state $s \in \mathcal{D}$ independently. Expanding the terms for a specific state s yields:

$$J(\pi_\theta(\cdot|s)) = \int \pi_\theta(a|s)R(s,a)da - \gamma \int \pi_\theta(a|s)\log\frac{\pi_\theta(a|s)}{\pi^{\text{old}}(a|s)}da.$$

To ensure $\pi_\theta(\cdot|s)$ is a valid probability density, a state-dependent Lagrangian multiplier $\lambda(s)$ is introduced for the normalization constraint $\int \pi_\theta(a|s)da = 1$, yielding the Lagrangian functional:

$$\begin{aligned} L(\pi_\theta, \lambda) = &\int \pi_\theta(a|s)R(s,a)da - \gamma \int \pi_\theta(a|s)\log\frac{\pi_\theta(a|s)}{\pi^{\text{old}}(a|s)}da \\ &+ \lambda(s)\left(1 - \int \pi_\theta(a|s)da\right). \end{aligned}$$

Taking the functional derivative of L with respect to the density $\pi_\theta(a|s)$ and setting it to zero yields the first-order optimality condition:

$$\frac{\delta L}{\delta \pi_\theta(a|s)} = R(s,a) - \gamma\left(\log\frac{\pi_\theta(a|s)}{\pi^{\text{old}}(a|s)} + 1\right) - \lambda(s) = 0.$$

Rearranging the terms to isolate $\pi_\theta(a|s)$:

$$\begin{aligned} \log\frac{\pi_\theta(a|s)}{\pi^{\text{old}}(a|s)} &= \frac{R(s,a)}{\gamma} - 1 - \frac{\lambda(s)}{\gamma}, \\ \pi_\theta(a|s) &= \pi^{\text{old}}(a|s)\exp\left(\frac{R(s,a)}{\gamma}\right)\exp\left(-1 - \frac{\lambda(s)}{\gamma}\right). \end{aligned}$$

Since the term $\exp\left(-1 - \frac{\lambda(s)}{\gamma}\right)$ is independent of a, it acts as a normalization constant. Defining its inverse as the partition function $Z(s)$:

$$Z(s) = \exp\left(1 + \frac{\lambda(s)}{\gamma}\right) = \int \pi^{\text{old}}(a|s)\exp\left(\frac{R(s,a)}{\gamma}\right)da.$$

Substituting $Z(s)$ back yields the theoretical optimal policy π^* corresponding to Eq. equation 6:

$$\pi^*(a|s) = \frac{1}{Z(s)}\pi^{\text{old}}(a|s)\exp\left(\frac{R(s,a)}{\gamma}\right). \tag{13}$$

A.2 Proof of Proposition 1

Proposition 1. *The optimal marginal density at time t relates to the reference density p_t^{old} via a time-dependent value function $\phi(x_t, c, t)$:*

$$p_t^*(x_t|c) = \frac{1}{Z(c)}p_t^{\text{old}}(x_t|c)\cdot\phi(x_t,c,t),$$

where $\phi(x_t,c,t) = \mathbb{E}_{x_1\sim p^{old}(\cdot|x_t,c)}\left[\exp\left(\frac{R(c,x_1)}{\gamma}\right)\right]$.

Proof. The marginal density $p_t^*(x_t|c)$ is obtained by marginalizing the joint distribution over the terminal state x_1:

$$p_t^*(x_t|c) = \int p(x_t|x_1)p_1^*(x_1|c)dx_1.$$

The forward transition kernel $p(x_t|x_1)$ is determined by the ReFlow schedule (e.g., $x_t = tx_1 + (1-t)x_0$) and remains strictly independent of the policy. Substituting the optimal terminal distribution $p_1^*(x_1|c) = \frac{1}{Z(c)}p_1^{\text{old}}(x_1|c)\exp\left(\frac{R(c,x_1)}{\gamma}\right)$:

$$p_t^*(x_t|c) = \frac{1}{Z(c)}\int p(x_t|x_1)p_1^{\text{old}}(x_1|c)\exp\left(\frac{R(c,x_1)}{\gamma}\right)dx_1.$$

Applying Bayes' theorem to the reference policy terms, the joint probability can be factorized as $p(x_t|x_1)p_1^{\text{old}}(x_1|c) = p^{\text{old}}(x_1|x_t,c)p_t^{\text{old}}(x_t|c)$. Substituting this factorization back into the integral formulation:

$$p_t^*(x_t|c) = \frac{1}{Z(c)}\int p_t^{\text{old}}(x_t|c)p^{\text{old}}(x_1|x_t,c)\exp\left(\frac{R(c,x_1)}{\gamma}\right)dx_1.$$

Since the intermediate reference marginal $p_t^{\text{old}}(x_t|c)$ is independent of the integration variable x_1, it can be factored out of the integral:

$$p_t^*(x_t|c) = \frac{1}{Z(c)}p_t^{\text{old}}(x_t|c)\underbrace{\int p^{\text{old}}(x_1|x_t,c)\exp\left(\frac{R(c,x_1)}{\gamma}\right)dx_1}_{\phi(x_t,c,t)}.$$

The integral term mathematically corresponds to the conditional expectation over the posterior distribution $p^{\text{old}}(x_1|x_t,c)$, which explicitly defines the value function $\phi(x_t,c,t)$. □

A.3 Derivation of Tweedie's Formula in Flow Matching

To establish the mathematical foundation, this section provides the analytical derivation of Tweedie's formula within the context of Flow Matching, which explicitly maps the posterior mean estimation to the marginal score function.

Under the ReFlow formulation, the forward probability path is constructed via a linear interpolation between the noise $x_0 \sim \mathcal{N}(0,I)$ and the data $x_1 \sim p_1$:

$$x_t = tx_1 + (1-t)x_0.$$

Given the Gaussian conditional distribution $p(x_t|x_1) = \mathcal{N}(x_t; tx_1, (1-t)^2 I)$, the corresponding conditional score function evaluates to:

$$\nabla_{x_t}\log p(x_t|x_1) = -\frac{x_t - tx_1}{(1-t)^2}.$$

The marginal score function $\nabla_{x_t}\log p_t(x_t|c)$ can be expressed by marginalizing over the terminal data distribution $p_1(x_1|c)$ and applying Bayes' theorem:

$$\begin{aligned}
\nabla_{x_t}\log p_t(x_t|c) &= \frac{1}{p_t(x_t|c)}\nabla_{x_t}p_t(x_t|c) \\
&= \frac{1}{p_t(x_t|c)}\int p_1(x_1|c)\nabla_{x_t}p(x_t|x_1)dx_1 \\
&= \int \frac{p(x_t|x_1)p_1(x_1|c)}{p_t(x_t|c)}\nabla_{x_t}\log p(x_t|x_1)dx_1 \\
&= \int p(x_1|x_t,c)\nabla_{x_t}\log p(x_t|x_1)dx_1.
\end{aligned}$$

Substituting the explicit form of the conditional score function into the integral gives:

$$\begin{aligned}\nabla_{x_t}\log p_t(x_t|c) &= \int p(x_1|x_t,c)\left(-\frac{x_t-tx_1}{(1-t)^2}\right)dx_1\\ &= -\frac{x_t}{(1-t)^2}+\frac{t}{(1-t)^2}\int p(x_1|x_t,c)x_1dx_1.\end{aligned}$$

Substituting the posterior mean $\hat{x}_1(x_t,c)=\int p(x_1|x_t,c)x_1dx_1$:

$$\nabla_{x_t}\log p_t(x_t|c) = -\frac{x_t}{(1-t)^2}+\frac{t}{(1-t)^2}\hat{x}_1(x_t,c).$$

Rearranging the terms to isolate $\hat{x}_1(x_t,c)$ yields Tweedie's formula, connecting the posterior mean estimation directly to the marginal score:

$$\hat{x}_1(x_t,c) = \frac{x_t}{t}+\frac{(1-t)^2}{t}\nabla_{x_t}\log p_t(x_t|c). \tag{14}$$

A.4 Proof of Theorem 1

Theorem 1. *The optimal velocity field $v^*(x_t,c,t)$ can be equivalently expressed as the reference field rectified by an advantage-weighted residual expectation:*

$$v^*(x_t,c,t) = v^{\text{old}}(x_t,c,t)+\mathbb{E}_{x_1\sim p^{\text{old}}(\cdot|x_t,c)}\left[\mathcal{A}(x_1,x_t)\left(u_t(x_t|x_1)-v^{\text{old}}(x_t,c,t)\right)\right].$$

where $\mathcal{A}(x_1,x_t)$ is the Centered Advantage, constructed by leveraging the zero-expectation property $\mathbb{E}[u_t-v^{old}]=0$ to subtract a constant baseline of 1:

$$\mathcal{A}(x_1,x_t) = \frac{\exp\left(\frac{R(c,x_1)}{\gamma}\right)}{\phi(x_t,c,t)}-1.$$

Proof. From Eq. equation 8, the optimal velocity field is strictly determined by the score function of the intermediate optimal marginal:

$$v^*(x_t,c,t) = v^{\text{old}}(x_t,c,t)+\frac{1-t}{t}\nabla_{x_t}\log\phi(x_t,c,t). \tag{15}$$

Expanding the gradient of the log value function and applying the logarithmic derivative identity $\nabla_{x_t}p = p\nabla_{x_t}\log p$, the gradient is derived as:

$$\begin{aligned}\nabla_{x_t}\log\phi(x_t,c,t) &= \frac{\nabla_{x_t}\phi(x_t,c,t)}{\phi(x_t,c,t)}\\ &= \frac{1}{\phi(x_t,c,t)}\nabla_{x_t}\int p^{\text{old}}(x_1|x_t,c)\exp\left(\frac{R(c,x_1)}{\gamma}\right)dx_1\\ &= \frac{1}{\phi(x_t,c,t)}\int p^{\text{old}}(x_1|x_t,c)\exp\left(\frac{R(c,x_1)}{\gamma}\right)\\ &\quad\cdot\nabla_{x_t}\log p^{\text{old}}(x_1|x_t,c)dx_1.\end{aligned} \tag{16}$$

Relating the score of the posterior $\nabla_{x_t}\log p^{\text{old}}(x_1|x_t,c)$ to the forward transition fields via Bayes' theorem factorizes the gradient into a direct linear relationship with the velocity residual:

$$\begin{aligned}\nabla_{x_t}\log p^{\text{old}}(x_1|x_t,c) &= \nabla_{x_t}\log p(x_t|x_1)-\nabla_{x_t}\log p_t^{\text{old}}(x_t|c)\\ &= \frac{tx_1-x_t}{(1-t)^2}-\frac{t\hat{x}_1^{\text{old}}-x_t}{(1-t)^2}\\ &= \frac{t}{1-t}\left(\frac{x_1-\hat{x}_1^{\text{old}}}{1-t}\right)\\ &= \frac{t}{1-t}\left(u_t(x_t|x_1)-v^{\text{old}}(x_t,c,t)\right),\end{aligned} \tag{17}$$

where the conditional and marginal scores are derived from Tweedie's formula in Sec A.3, while $u_t(x_t|x_1) = \frac{x_1 - x_t}{1-t}$ and $v^{\text{old}}(x_t, c, t) = \frac{\hat{x}_1^{\text{old}} - x_t}{1-t}$ denote the conditional flow target and reference velocity, respectively.

Substituting Eq.equation 17 into Eq.equation 16 transforms the integral into an expectation over the posterior:

$$\nabla_{x_t} \log \phi(x_t, c, t) = \frac{t}{1-t} \mathbb{E}_{x_1 \sim p^{\text{old}}(\cdot|x_t, c)} \Bigg[\frac{\exp\left(\frac{R(c,x_1)}{\gamma}\right)}{\phi(x_t, c, t)} \left(u_t(x_t|x_1) - v^{\text{old}}(x_t, c, t)\right) \Bigg].$$

Substituting this explicit gradient form back into the base velocity Eq.equation 15:

$$v^*(x_t, c, t) = v^{\text{old}}(x_t, c, t) + \mathbb{E}_{x_1 \sim p^{\text{old}}(\cdot|x_t, c)} \left[\frac{\exp\left(\frac{R(c,x_1)}{\gamma}\right)}{\phi(x_t, c, t)} \left(u_t(x_t|x_1) - v^{\text{old}}(x_t, c, t)\right) \right].$$

Since the velocity residual has zero expectation, $\mathbb{E}_{x_1 \sim p^{\text{old}}(\cdot|x_t, c)}[u_t(x_t|x_1) - v^{\text{old}}(x_t, c, t)] = 0$, subtracting a unit baseline from the weight preserves the equality:

$$v^*(x_t, c, t) = v^{\text{old}}(x_t, c, t) + \mathbb{E}_{x_1 \sim p^{\text{old}}(\cdot|x_t, c)} \left[\left(\frac{\exp\left(\frac{R(c,x_1)}{\gamma}\right)}{\phi(x_t, c, t)} - 1 \right) \left(u_t(x_t|x_1) - v^{\text{old}}(x_t, c, t)\right) \right].$$

Defining the centered advantage as $\mathcal{A}(x_1, x_t) = \frac{\exp(R(c,x_1)/\gamma)}{\phi(x_t,c,t)} - 1$ explicitly yields the exact formulation of the optimal velocity field, completing the proof. □

A.5 Proof of Theorem 2

Theorem 2. *The gradients of the Advantage-Weighted Rectification Loss $\mathcal{L}_{AWR}$ and the Ideal Loss $\mathcal{L}_{Ideal}$ are identical with respect to θ:*

$$\nabla_\theta \mathcal{L}_{\text{AWR}}(\theta) = \nabla_\theta \mathcal{L}_{\text{Ideal}}(\theta).$$

Proof. Define the stochastic target derived from Theorem 1 as $Y(x_1, x_t, c, t) = v^{\text{old}}(x_t, c, t) + \mathcal{A}(x_1, x_t)\left(u_t(x_t|x_1) - v^{\text{old}}(x_t, c, t)\right)$. Theorem 1 establishes that the theoretical optimal velocity field is the conditional expectation of this stochastic target: $v^*(x_t, c, t) = \mathbb{E}_{x_1 \sim p^{\text{old}}(\cdot|x_t, c)}[Y(x_1, x_t, c, t)]$.

The Advantage-Weighted Rectification loss is formulated as:

$$\mathcal{L}_{\text{AWR}}(\theta) = \mathbb{E}_{t,c,x_t}\left[\mathbb{E}_{x_1 \sim p^{\text{old}}(\cdot|x_t, c)}\left[\|v_\theta(x_t, c, t) - Y(x_1, x_t, c, t)\|^2\right]\right].$$

Expanding the squared L_2 norm by introducing the optimal field $v^*(x_t, c, t)$ yields a standard bias-variance decomposition:

$$\begin{aligned}\|v_\theta - Y\|^2 &= \|(v_\theta - v^*) + (v^* - Y)\|^2 \\ &= \|v_\theta - v^*\|^2 + \|v^* - Y\|^2 + 2\langle v_\theta - v^*, v^* - Y\rangle.\end{aligned}$$

Substituting this decomposition into the expectation separates the loss function into three distinct terms:

$$\begin{aligned}\mathcal{L}_{\text{AWR}}(\theta) = &\mathbb{E}_{t,c,x_t}\left[\|v_\theta(x_t, c, t) - v^*(x_t, c, t)\|^2\right] \\ &+ \mathbb{E}_{t,c,x_t,x_1}\left[\|v^*(x_t, c, t) - Y(x_1, x_t, c, t)\|^2\right] \\ &+ \mathbb{E}_{t,c,x_t}\left[2\langle v_\theta - v^*, \mathbb{E}_{x_1|x_t,c}[v^* - Y]\rangle\right].\end{aligned}$$

The three terms evaluate as follows:

- The first term is strictly the Ideal Loss $\mathcal{L}_{\text{Ideal}}(\theta)$, which measures the regression error between the parameterized velocity and the theoretical optimal field.
- The second term represents the irreducible variance of the stochastic target Y. Because this term depends solely on the data and the reference policy, it is independent of the model parameters θ, yielding a gradient of zero.
- In the third term, the inner expectation evaluates to $\mathbb{E}_{x_1|x_t,c}[v^* - Y] = v^* - \mathbb{E}_{x_1|x_t,c}[Y]$. By the definition of v^*, this equals $v^* - v^* = 0$, causing the entire cross-term to vanish.

Consequently, the AWR loss simplifies to the Ideal Loss plus an independent constant term C:

$$\mathcal{L}_{\text{AWR}}(\theta) = \mathcal{L}_{\text{Ideal}}(\theta) + C.$$

Taking the gradient with respect to θ eliminates the constant C, proving the exact optimization equivalence:

$$\nabla_\theta \mathcal{L}_{\text{AWR}}(\theta) = \nabla_\theta \mathcal{L}_{\text{Ideal}}(\theta).$$

□

B Theoretical Discussions with RWR and DPO

RWR and DPO represent distinct alignment paradigms that originate from the same theoretical optimal policy. Despite their divergent practical implementations, both methods can be mathematically unified under the KL-constrained reward maximization framework, leveraging the closed-form optimal density derived in Eq.equation 13.

B.1 Connection to RWR in Flow Matching

Under the ReFlow schedule $x_t = tx_1 + (1-t)x_0$, aligning a parameterized velocity $v_\theta(x_t, c, t)$ with the ideal marginal velocity $v^*(x_t, c, t) = \mathbb{E}_{x_1 \sim p^*(\cdot|x_t,c)}[u_t(x_t|x_1)]$ yields the global matching objective:

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{t, x_t \sim p_t^*}\left[\|v_\theta(x_t, c, t) - v^*(x_t, c, t)\|^2\right].$$

To bypass the intractable marginal expectation in v^*, Conditional Flow Matching (CFM) establishes a gradient equivalence. By expanding the squared norm and marginalizing the cross-term $\mathbb{E}_{x_t \sim p_t^*}[\langle v_\theta, v^* \rangle]$, the objective reduces to matching the conditional field:

$$\iint p_1^*(x_1|c)p(x_t|x_1)\langle v_\theta, u_t\rangle dx_1 dx_t = \mathbb{E}_{x_1 \sim p_1^*}\mathbb{E}_{x_t \sim p(\cdot|x_1)}[\langle v_\theta, u_t\rangle].$$

which recovers the conditional matching objective over the optimal target distribution:

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t, x_1 \sim p_1^*, x_0 \sim p_0}\left[\|v_\theta(x_t, c, t) - (x_1 - x_0)\|^2\right].$$

Since sampling directly from p_1^* is practically infeasible, applying importance sampling from the accessible reference policy p_1^{old} re-weights the expectation by the density ratio $\frac{p_1^*(x_1|c)}{p_1^{\text{old}}(x_1|c)} \propto \exp\left(\frac{R(c,x_1)}{\gamma}\right)$. Absorbing the partition function $Z(c)$ establishes the final RWR objective:

$$\mathcal{L}_{\text{RWR}}(\theta) \propto \mathbb{E}_{x_1 \sim p_1^{\text{old}}}\left[\exp\left(\frac{R(c, x_1)}{\gamma}\right)\mathbb{E}_{t, x_0 \sim p_0}\left[\|v_\theta(x_t, c, t) - (x_1 - x_0)\|^2\right]\right].$$

which bridges the classical RWR alignment paradigm with continuous generative processes, demonstrating that exponentiated reward re-weighting naturally emerges from projecting the velocity onto the optimal policy trajectory.

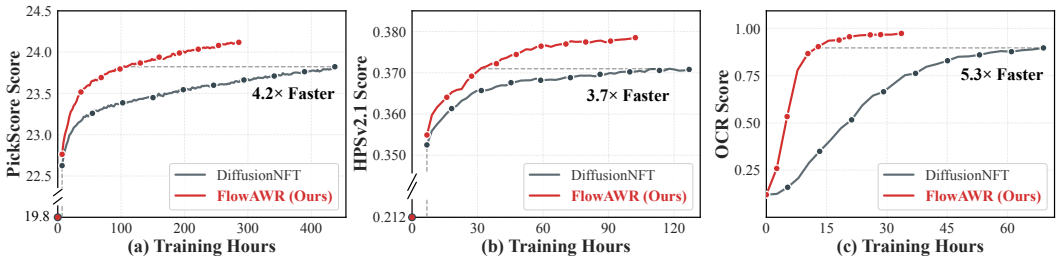


Figure 6: FlowAWR versus DiffusionNFT on (a) PickScore, (b) HPSv2.1, and (c) OCR, against GPU-hours. Dashed horizontal lines compare the cost of reaching DiffusionNFT's highest recorded reward under the plotted configuration.

B.2 Connection to DPO

Unlike RWR's regression approach, DPO bypasses explicit reward modeling. Rearranging the logarithmic form of the optimal policy analytically reparameterizes the unobserved reward $R(s, a)$ strictly in terms of policy probabilities:

$$R(s,a) = \gamma \log \frac{\pi^*(a|s)}{\pi^{\text{old}}(a|s)} + \gamma \log Z(s).$$

Under the Bradley-Terry (BT) preference model, the probability that an action y_w is preferred over y_l is defined by the logistic function of their reward difference, $P(y_w \succ y_l|s) = \sigma\left(R(s, y_w) - R(s, y_l)\right)$. Substituting the reparameterized rewards perfectly cancels the intractable partition function $Z(s)$:

$$R(s,y_w) - R(s,y_l) = \gamma \log \frac{\pi^*(y_w|s)}{\pi^{\text{old}}(y_w|s)} - \gamma \log \frac{\pi^*(y_l|s)}{\pi^{\text{old}}(y_l|s)}.$$

Parameterizing π^* with a learnable policy π_θ and minimizing the negative log-likelihood over a preference dataset $\mathcal{D}$ directly yields the exact DPO objective:

$$\mathcal{L}_{\text{DPO}}(\theta) = -\mathbb{E}_{(s,y_w,y_l)\sim\mathcal{D}}\left[\log \sigma\left(\gamma \log \frac{\pi_\theta(y_w|s)}{\pi^{\text{old}}(y_w|s)} - \gamma \log \frac{\pi_\theta(y_l|s)}{\pi^{\text{old}}(y_l|s)}\right)\right]. \tag{18}$$

which confirms that DPO structurally inherits the KL-constrained optimality criteria, simply reformulating it to accommodate pairwise preference data.

C Experiment Details

C.1 Training Cost and Additional Baseline Comparisons

We report sampling-and-training cost in GPU-hours. The horizontal axes in Figures 6 and 7, labeled "Training Hours," are computed by multiplying each method's per-update time by its update count and eight GPUs, then dividing by 3,600. DiffusionNFT uses its own timing measurements in both figures. The FlowAWR–DiffusionNFT comparison concerns single-reward training with 10-step rollouts and 40-step ODE evaluation.

Computational Cost. We report aggregate GPU-hours across eight GPUs, including sampling costs amortized over the associated training updates. For TempFlow-GRPO, we additionally account for 70 seconds of reward computation per training update before aggregation across GPUs. On SD3.5-Medium, FlowAWR requires 0.2549, 0.1292, 0.2285, and 0.2273 GPU-hours per update on GenEval, OCR, PickScore, and HPSv2.1, respectively. Compared with DiffusionNFT, FlowAWR reduces the per-update cost by 2.5% on OCR, while the differences on the other three tasks remain below 0.4%. Relative to AWM, FlowAWR reduces the cost by 35.8%, 31.6%, and 1.9% on GenEval, OCR, and PickScore, respectively, while incurring a 0.4% increase on HPSv2.1. Compared with TempFlow-GRPO, the corresponding reductions are 52.1%, 82.2%, 56.7%, and 57.7%. For FLUX.1-dev on HPSv3, we instead report the cost of a complete sampling iteration, including all

subsequent training updates: one update for FlowAWR and two for TempFlow-GRPO. Under this accounting, FlowAWR requires 1.5909 GPU-hours per sampling iteration, compared with 3.1422 for TempFlow-GRPO, a reduction of 49.4%.

Figure 6 compares FlowAWR with DiffusionNFT. Its speedup annotations use DiffusionNFT's cost to reach its highest recorded reward divided by FlowAWR's cost at the first evaluated checkpoint reaching that reward. The OCR curves use DiffusionNFT's general configuration, rather than the OCR-specific configuration in Table 1; Appendix C.5 documents this distinction.

Additional Baselines. We further compare FlowAWR with AWM (Xue et al., 2025a), DiffusionNFT (Zheng et al., 2025), and TempFlow-GRPO (He et al., 2026) on three single-reward optimization tasks: PickScore, HPSv2.1, and OCR. Figure 7 reports their learning curves against training cost measured in GPU-hours.

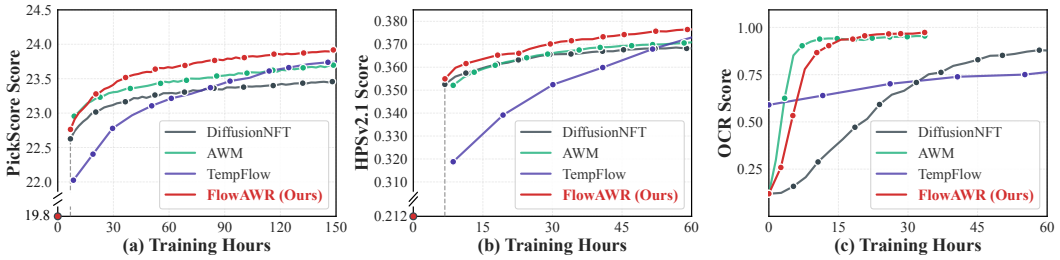


Figure 7: Single-reward learning curves for FlowAWR, AWM, DiffusionNFT, and TempFlow-GRPO on (a) PickScore, (b) HPSv2.1, and (c) OCR. Training cost is measured in GPU-hours.

C.2 Experiments on FLUX.1-dev

We further evaluate FlowAWR on FLUX.1-dev against DiffusionNFT and TempFlow-GRPO using the HPDv2 dataset, consistent with TempFlow-GRPO (He et al., 2026). All methods use a training resolution of 720×720 and an evaluation resolution of 1024×1024. Following the rollout-only CFG strategy of Xu et al. (2026), we enable CFG during rollout sampling but disable it when computing the training loss. CFG is also enabled during evaluation. Figure 8 compares HPSv3 (Ma et al., 2025) learning curves against GPU-hours. FlowAWR reaches a higher HPSv3 score than both baselines within the displayed training range.

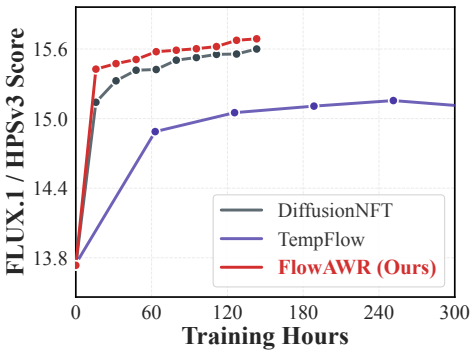
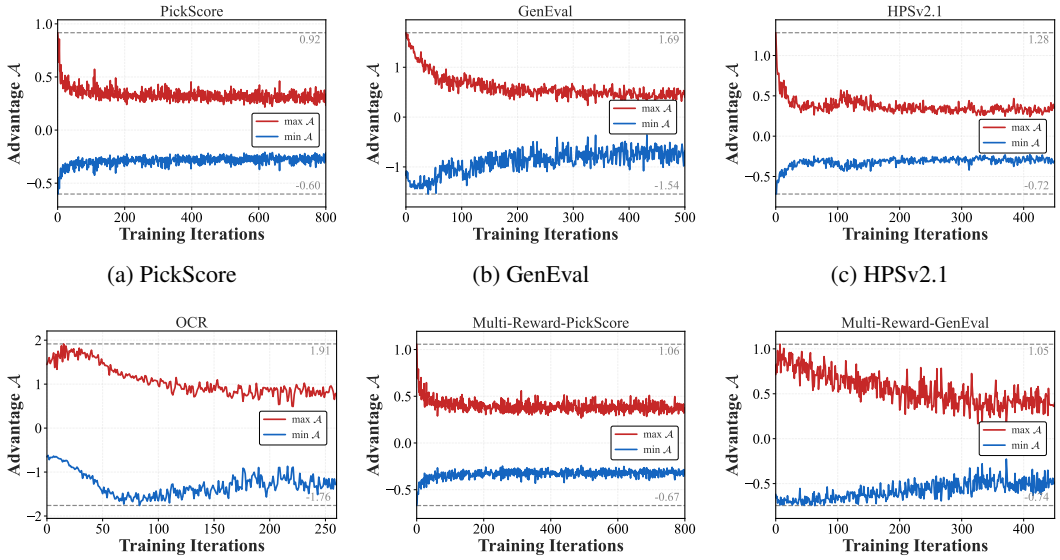


Figure 8: HPSv3 learning curves for FlowAWR, DiffusionNFT, and TempFlow-GRPO on FLUX.1-dev using HPDv2. Training cost is measured in GPU-hours.

C.3 Empirical Range of the Centered Advantage

To validate the numerical stability of the proposed framework, Figure 9 illustrates the empirical variation range of the centered advantage $\mathcal{A}(x_1, x_t)$ across six distinct alignment tasks, encompassing both single-reward (PickScore, GenEval, HPSv2.1, OCR) and multi-reward optimizations. While the advantage values exhibit the necessary fluctuations to provide discriminative learning signals, their overall dynamic range remains strictly bounded and controllable throughout the training process. The effective coefficients remain bounded under the task-specific scaling, which allows values below -1 for GenEval and OCR. These results illustrate how reward normalization and task-specific scaling control coefficient magnitudes across tasks without directly clipping the advantages.



(a) PickScore (b) GenEval (c) HPSv2.1

(d) OCR (e) Multi-Reward (PickScore) (f) Multi-Reward (GenEval)

Figure 9: Empirical variation of the centered advantage $\mathcal{A}(x_1, x_t)$ across six alignment tasks. Under both single-reward and multi-reward settings, $\mathcal{A}(x_1, x_t)$ remains structurally bounded. It provides discriminative learning signals while naturally restricting the upper and lower bound.

C.4 Experimental Setup

Training Configurations. The training setup follows FlowGRPO and DiffusionNFT, which adopts 48 groups per epoch, a group size of 24, and a LoRA parameterization ($r = 32$, $\alpha = 64$) with a constant learning rate of 3×10^{-4}. For each collected clean image, forward noising and loss computation are performed on the corresponding sampling timesteps. Data collection employs a 2nd-order ODE sampler with adaptive time weighting by default.

Single-Reward Optimization. For fair comparison with DiffusionNFT, we use the same 10 sampling steps and the shared EMA decay schedule $\mu_i = \min(0.001i, 0.5)$ across the single-reward tasks, while DiffusionNFT uses its reported configurations. HPSv2.1 optimization uses the same Pick-a-Pic dataset as PickScore.

Advantage Scaling. We compute within-prompt softmax probabilities p_i from reward logits clipped to $[-5, 5]$, using max-subtraction and float64 arithmetic. The raw advantage is $\mathcal{A}_i^{\text{raw}} = p_i - 1/G$. PickScore and HPSv2.1 use $G\mathcal{A}_i^{\text{raw}}$, following Eq. equation 11 with clipped logits. For rule-based rewards such as GenEval and OCR, closely clustered rewards within a prompt group can yield small group-relative advantages. We therefore rescale the centered advantages by their within-group standard deviation to adapt the correction magnitude, using a lower-bounded denominator to avoid excessive amplification.

Single-Branch Advantage Ablation. The fixed-coefficient variants in Figure 5(d,e) use $\beta \in \{1, 0.1\}$ and select one NFT-style target according to the sign of each sample's advantage. Positive advantages select $v^{\text{old}} + (u_t - v^{\text{old}})/\beta$ with outer loss weight r, while negative advantages select $v^{\text{old}} - (u_t - v^{\text{old}})/\beta$ with weight $1 - r$, where r is the normalized reward used in the NFT loss. Each sample contributes only its selected branch. In additional trials, omitting these outer reward weights also led to reward collapse.

Multi-Reward Optimization. Following the multi-stage curriculum of DiffusionNFT, the training is structured around three dataset-specific configurations: (1) **3 rewards** (PickScore, CLIPScore, and HPSv2.1) on the Pick-a-Pic dataset; (2) **4 rewards** (the previous three plus GenEval) on the GenEval dataset; and (3) **4 rewards** (the initial three plus OCR) on the OCR dataset. To overcome the low quality of initial CFG-free generations, the curriculum sequentially applies configuration (1) for 800 iterations, (2) for 200 iterations, (1) for 200 iterations, (2) for 200 iterations, and finally

concludes with (3) for 100 iterations. Within each stage, all active rewards are equally weighted, with PickScore explicitly divided by 26 for $[0, 1]$ normalization. For data collection, we maintain the sampling steps as 10 to ensure training efficiency. We maintain the shared EMA decay schedule $\mu_i = \min(0.001i, 0.5)$ across all stages.

Note that these iteration counts are approximate. To ensure continuous optimization during stage transitions, the complete training state is strictly resumed from the preceding stage, including the model parameters θ, the reference policy θ_{old}, the optimizer, and the gradient scaler.

C.5 DiffusionNFT OCR Configurations

The OCR checkpoint in Table 1 uses DiffusionNFT's official task-specific configuration, whereas the OCR learning curves in Figures 3(c), 6(c), and 7(c) use its general GenEval/PickScore configuration. For DiffusionNFT, η_i controls the EMA update of the lagging old policy:

$$\theta_i^{\text{old}} \leftarrow \eta_i \theta_{i-1}^{\text{old}} + (1 - \eta_i)\theta_i.$$

The general configuration uses $\beta = 1$ and $\eta_i = \min(0.001i, 0.5)$. The OCR-specific configuration jointly sets $\beta = 0.1$ and uses the delayed schedule

$$\eta_i = \begin{cases} 0, & i < 75, \\ \min(0.0075(i - 75), 0.999), & i \geq 75. \end{cases}$$

FlowAWR retains the shared EMA schedule specified in Section 4.1, without OCR-specific adjustment.

Table S1 compares both DiffusionNFT configurations with FlowAWR, which achieves an OCR score of 0.975 after 260 updates. The general configuration obtains an OCR score of 0.89 after 520 updates, compared with 0.98 after 400 updates for the OCR-specific configuration. This comparison indicates sensitivity to the jointly selected (β, η_i) configuration, but does not isolate either parameter's individual contribution or their interaction. It compares separate training configurations with different iteration budgets, not a decline within one training run; the non-OCR metrics do not uniformly decline.

Table S1: Comparison of FlowAWR with DiffusionNFT under general and OCR-specific configurations for single-reward OCR training. The OCR-specific DiffusionNFT and FlowAWR results are reproduced from Table 1. All runs use 10-step training rollouts and 40-step first-order ODE evaluation.

Method / Configuration	#Iter	Task Metric			DrawBench Metric				
		GenEval	OCR	PickScore	CLIPScore	HPSv2.1	Aesthetic	ImgRwd	PickScore
DiffusionNFT (General)	0.52k	0.24	0.89	19.38	0.240	0.183	4.90	-0.92	20.27
DiffusionNFT (OCR-specific)	0.4k	0.20	0.98	19.15	0.233	0.168	4.79	-1.27	20.15
FlowAWR	**0.26k**	0.29	0.975	19.43	0.246	0.193	5.00	-0.78	20.51

D Additional Qualitative Results

We provide additional qualitative comparisons between the base model, FlowGRPO, DiffusionNFT, and our proposed method in Figure 10, Figure 11, and Figure 12, corresponding to the GenEval, OCR, and DrawBench benchmarks, respectively.



Figure 10: Qualitative comparison on the GenEval benchmark.



A vibrant urban wall features bold, dynamic graffiti spelling "Revolution Now" in striking colors, capturing the energy and urgency of a movement.

A vibrant hot air balloon ascends into a clear blue sky, trailing a banner that reads "Adventure Awaits" in bold, flowing letters.

A bustling train station with a vintage aesthetic, the platform speaker hanging from a metal pole, clearly announcing "Now Boarding Track 9" amidst the crowd of travelers …

A realistic movie set with a clapperboard slate clearly marked "Take 3", held by a production assistant in front of a vintage film camera, under the warm glow of studio lights.

A treehouse nestled among the branches, with a weathered sign nailed to the trunk, clearly displaying "No Adults Allowed" in bold, playful letters.

Figure 11: Qualitative comparison on OCR tasks.

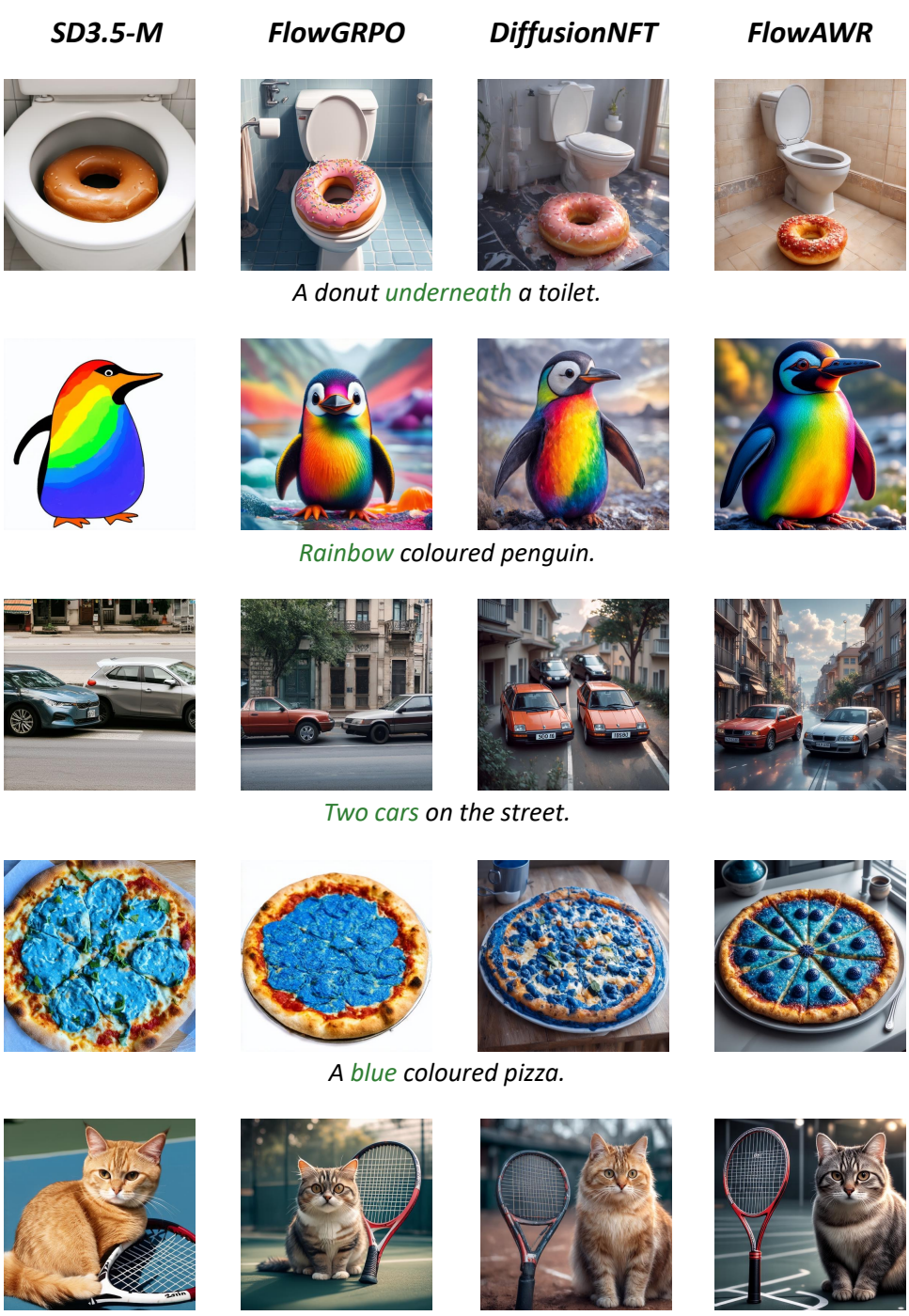


Figure 12: Qualitative comparison on the DrawBench benchmark.