EXPLICIT CRITIC GUIDANCE FOR ALIGNING DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Online reinforcement learning is becoming increasingly important for aligning diffusion models with non-differentiable objectives. However, existing methods still face limitations in assigning fine-grained credit along denoising trajectories and in realizing stable value-based optimization. We propose a state-aligned latent actor-critic framework for diffusion post-training, in which the diffusion model serves as its own timestep-conditioned value function and predicts values directly on noisy latent states. This enables trajectory-level PPO training, supports stable actor-critic optimization with simple conditioning and value pretraining strategies, and naturally allows the learned critic to be reused for inference-time steering. We further extend the framework to multi-reward optimization, using reward-specific value heads to support complementary objectives. Across both UNet- and DiT-based backbones, our method improves single-reward performance over DDPO and DanceGRPO in our controlled comparisons and supports multi-reward optimization with objective-dependent trade-offs, while test-time steering provides additional gains in generation quality.

1 INTRODUCTION

Post-training is becoming an increasingly important stage for visual generative models. Although large-scale pretraining (Esser et al., 2024, Seawead et al., 2025, Wu et al., 2025a, Liang et al., 2024, Yin et al., 2025) already yields strong visual quality, pretrained models often remain suboptimal on properties that users directly care about, including prompt faithfulness, compositional correctness (Ghosh et al., 2023, Huang et al., 2023), text rendering (Cui et al., 2025), aesthetics, and human preference (Wu et al., 2023c, Ma et al., 2025, Xu et al., 2023, Kirstain et al., 2023, Wang et al., 2025). Reinforcement learning (Schulman et al., 2017, Ahmadian et al., 2024, Li et al., 2023, Shao et al., 2024, Zheng et al., 2025, Schulman et al., 2015, Hu et al., 2025, Ouyang et al., 2022, Stiennon et al., 2020, Bai et al., 2022a) is particularly appealing in this setting because it enables optimization with respect to verifiable and task-specific objectives, including learned preference models and rule-based rewards.

A central challenge in diffusion post-training is *credit assignment*. Diffusion models (Ho et al., 2020, Song & Ermon, 2019, Song et al., 2020, Dhariwal & Nichol, 2021, Lipman et al., 2023, Liu et al., 2023) generate samples through multi-step denoising trajectories, whereas rewards are typically defined only on the final decoded image. The problem is therefore not only to maximize terminal reward, but to assign it to the intermediate noisy states and denoising decisions that produced it. Group-relative methods (Liu et al., 2025b, Xue et al., 2025b, Deng et al., 2026) compare terminal rewards within a group, but apply the same trajectory-level advantage across denoising steps, without a state-dependent reference. Our state-ranking experiment illustrates the limitation of relying on a single terminal outcome: at early denoising steps, final rewards recover the ordering of expected state values with only 58.8% accuracy, close to random guessing. A learned critic raises this accuracy to 69.0%, providing a more informative state-dependent reference.

The next question is what representation such a critic should use. Existing diffusion critics (Zhao et al., 2025, McAllister et al., 2025) are typically defined in pixel space: they evaluate reconstructed clean-image proxies rather than the actual noisy latent states visited by the policy. Evaluating these proxies requires repeated image reconstruction and decoding, and their representations may not capture the expected return from a noisy state. A useful critic must therefore be assessed by its ability

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: **Text-conditioned image generation results from our SD3.5M models.** All samples are generated **without** classifier free guidance.

to predict continuation values and distinguish promising continuations, in addition to its effect on final policy performance.

To address these issues, we propose a *state-aligned latent actor-critic* framework in which a separate pretrained diffusion backbone serves as a timestep-conditioned value network and predicts $V_\phi(z_t, t, y)$ directly on noisy latent states. This requires only lightweight modifications to the pretrained diffusion backbone, allowing the critic to inherit the model's generative prior while operating on the same state space and diffusion-native features as the policy. The resulting critic is naturally noise-aware, avoids repeated image decoding, and can be reused at inference time, where its latent-space gradients provide a direct steering signal during sampling. To make this formulation practical, we further identify timestep-aware conditioning and a short value pretraining stage as simple but effective techniques for improving training stability and efficiency. As illustrated in Fig. 1, the framework also enables *CFG-free* post-training, reducing memory overhead and training cost while avoiding the color saturation artifacts and instability often introduced by classifier-free guidance.

Our experiments address these two questions directly. The diffusion critic improves state-ranking accuracy at early, middle, and late denoising stages. Compared with trained CLIP and HPS critics, it also achieves lower test-set value MSE and higher action-ranking accuracy (78.5% versus 74.2% and 64.0%). These predictive gains accompany improved single-reward performance over DDPO and DanceGRPO in our controlled comparisons on SD1.5 and SD3.5M. We further evaluate multi-reward optimization and inference-time steering in Appendix K, with a discussion of proxy overoptimization in Appendix J.

In summary, our contributions are as follows:

- We propose a state-aligned latent critic framework for diffusion post-training, which adapts a pretrained diffusion backbone into a timestep-aware value network with lightweight modifications and supports both latent-state value prediction and inference-time steering.
- We provide empirical evidence that a learned state-dependent critic improves intermediate state ranking, and that diffusion features yield more accurate value predictions and action rankings than the tested CLIP and HPS critics.
- We develop a practical training recipe with timestep-aware conditioning and value pretraining, and extend the learned critic to multiple reward objectives and inference-time steering.
- On SD1.5 and SD3.5M, our method improves single-reward performance over DDPO and DanceGRPO in our controlled comparisons, supports multi-reward optimization, and provides additional gains through inference-time steering.

2 Related Work

Diffusion alignment. Recent work on aligning diffusion models can be broadly grouped into reward-based fine-tuning, preference optimization, and online reinforcement learning. Early approaches align text-to-image models with human or learned feedback by collecting preference data, training reward models, and fine-tuning the generator with reward-based objectives (Lee et al., 2023, Fan et al., 2025a, Peng et al., 2019, Xue et al., 2025a). For differentiable rewards, alignment can also be performed by backpropagating reward gradients through the denoising process (Xu et al., 2023, Clark et al., 2024, Prabhudesai et al., 2023, 2024). Another line of work adapts preference optimization methods to diffusion models, including DPO-style formulations (Rafailov et al., 2023, Wu et al., 2025b, Liu et al., 2025c, Liang et al., 2025, Li et al., 2024, Zhang et al., 2025b, 2024a, Furuta et al., 2024, Yang et al., 2024a). More recently, online RL methods (Black et al., 2024, Fan et al., 2023, Schulman et al., 2017, Gupta et al., 2025, Miao et al., 2024, Zhao et al., 2025, McAllister et al., 2025, Zhang et al., 2024b) optimize diffusion models directly against downstream rewards. This line has also been extended to Group-Relative style policy gradient method (Liu et al., 2025b, Xue et al., 2025b, Li et al., 2025b,a), which bypass the training of critic. Compared with these methods, our work focuses on value-based post-training with an explicit critic learned on the same noisy latent trajectory as the policy. We provide extended discussion in Appendix E.

3 Method

3.1 Preliminaries

Denoising as a sequential decision process. We briefly define the sequential decision-making view of diffusion post-training used throughout the paper. Let z_t denote the noisy latent at timestep t, and let y be the text condition. A latent diffusion or flow-based generator produces an image by iteratively updating z_t along a denoising trajectory from noise to data. We view this process as a finite-horizon sequential decision problem. At step k, the state is

$$s_k = (z_{t_k}, t_k, y),$$

and the action a_k is the model prediction used by the sampler, e.g., a noise, score, or velocity prediction depending on the backbone. The transition is induced by the sampler. After the final step, the denoised latent is decoded and evaluated by a reward function, yielding a terminal reward $r_T = r(x, y)$, with $r_k = 0$ for $k < T$. The objective is to maximize the expected terminal reward over the denoising trajectory.

Group-relative policy optimization. GRPO-style diffusion post-training samples a group of G trajectories for a prompt and compares their terminal rewards. In its standard terminal-reward form, trajectory i receives

$$A_i^{\text{group}} = \frac{R_i - \mu_y}{\sigma_y + \varepsilon}, \qquad \mu_y = \frac{1}{G}\sum_{j=1}^{G} R_j, \tag{1}$$

where σ_y is the within-group reward standard deviation and $\varepsilon > 0$ ensures numerical stability. The same scalar weights the policy-gradient terms at all denoising steps of a trajectory. Here R_i is the terminal reward of trajectory i.

3.2 The Credit Assignment Limitation of GRPO

GRPO assigns the same group-relative advantage to every transition in a trajectory (Eq. (1)). However, a high final reward can reflect either a helpful transition or a state that was already promising. A common prompt-level baseline does not distinguish these cases: it does not account for what was expected from the particular state before the transition.

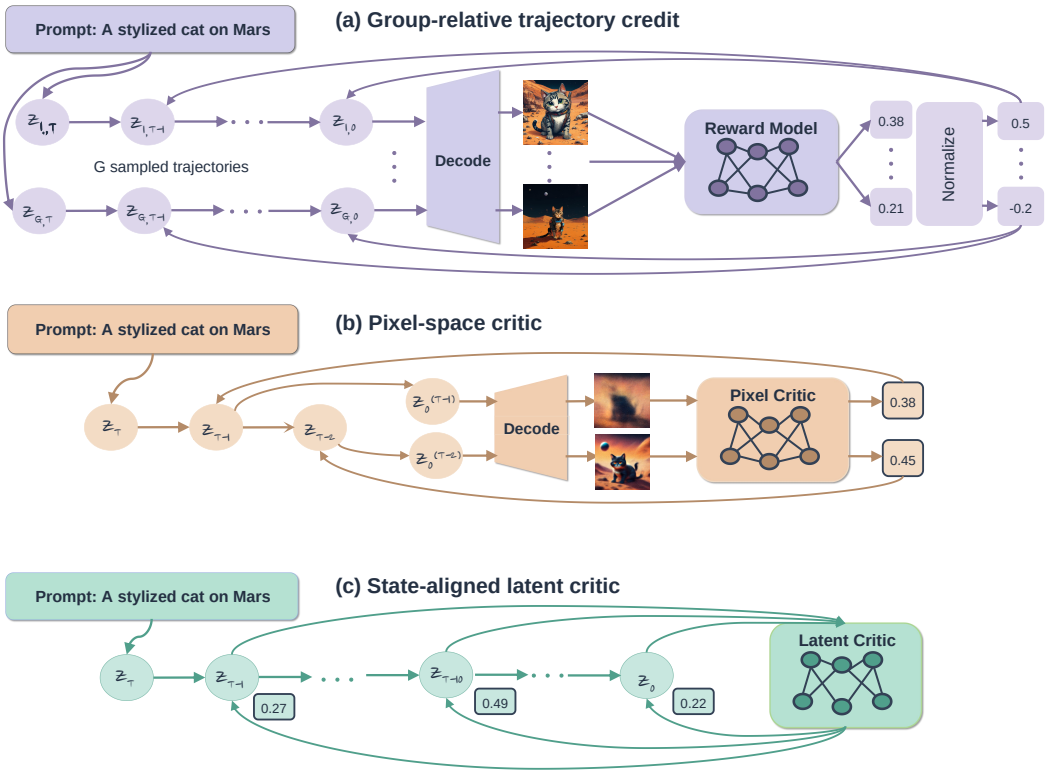

Figure 2: **Illustration of three credit assignment strategies in diffusion RL.** GRPO-style methods assign one relative signal to each trajectory by groupwise reward comparison. Pixel-space critics score reconstructed images decoded from intermediate latents, leading to proxy-state evaluation. Our method instead learns a critic directly on noisy latent states, enabling state-dependent and state-aligned credit assignment as well as inference-time steering from the same learned critic.

We examine how well terminal rewards reflect intermediate state quality using a frozen SD3.5M policy with HPSv2.1 reward. Across 200 prompts, we sample four trajectories each and retain states at steps 8, 20, and 32 of a 40-step stochastic sampler. Independent continuations estimate the expected return from each saved state. Within each prompt and stage, we compare the state ordering induced by final rewards with this continuation-based reference. GRPO preserves the final-reward ordering within a prompt group.

Table 1: State-ranking accuracy (%) on SD3.5M with HPSv2.1. Rankings from final rewards and the trained critic are compared with continuation-based state values.

Stage	Final reward	Critic
Early	58.8	**69.0**
Middle	87.5	**91.0**
Late	92.7	**94.0**

Table 1 shows that final-reward rankings agree with reference state-value rankings in only 58.8% of early-stage pairs, close to the 50% random-guessing baseline. Agreement rises to 87.5% and 92.7% at the middle and late stages. Thus, an individual terminal outcome is a particularly weak indicator of expected return early in denoising. The trained critic improves state-ranking accuracy at all three stages, as discussed in Sec. 4.1.

Together with the common-baseline limitation above, this motivates learning a reference conditioned on the current state:

$$V^{\pi}(s_k) = \mathbb{E}_{\pi}[R \mid s_k], \qquad V_{\phi}(s_k) = V_{\phi}(z_{t_k}, t_k, y) \approx V^{\pi}(s_k). \tag{2}$$

Centering the realized return as $R - V_\phi(s_k)$ calibrates each update against the expected outcome from its current state while preserving the terminal-reward objective. We next construct this critic directly on noisy diffusion latents.

3.3 Diffusion Models Are Timestep-Aware Critics

The critic must predict a future outcome from a partially denoised latent, its noise level, and the prompt. A pretrained diffusion backbone already processes precisely these inputs and learns features for continuing the denoising process. We use this representation as the starting point for value learning and train the scalar readout against terminal returns from sampled continuations.

We initialize a separate critic from the pretrained diffusion backbone used to initialize the actor and add a lightweight scalar readout. The actor and critic operate on the same state representation, with distinct trainable parameters θ and ϕ. Critic evaluation consumes (z_t, t, y) directly, avoiding image reconstruction and VAE decoding at each intermediate step. Final images are still decoded for terminal reward evaluation. Figure 3 presents the two backbone variants.

UNet critic. For a UNet, we patchify the output feature map into tokens, append a learnable summary token, and apply a lightweight attention head. A small MLP maps the resulting summary representation to a scalar.

DiT critic. For a diffusion transformer (Peebles & Xie, 2023), we attach critic branches at selected intermediate layers, following the use of intermediate diffusion features in Lin et al. (2025). Each branch uses a learnable query to attend to the visual tokens. The resulting critic tokens are concatenated and passed through an MLP to predict $V_\phi(z_t, t, y)$. Both variants aggregate spatial denoising features into a prediction of terminal return.

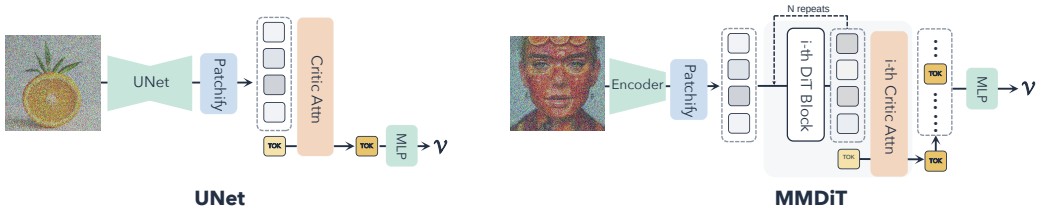

Figure 3: **Architecture overview over UNet and Diffusion Transformer.** We initialize the value network with pretrained diffusion models. Attention layer and MLP are added to predict critic scalar.

Because the information available about the final image changes throughout denoising, we condition the critic readout explicitly on t, in addition to the time conditioning already present in the backbone. An AdaLN-style module maps the timestep embedding to feature-wise scale, shift, and gate coefficients in the value readout. This allows the critic to interpret features differently at different noise levels. The backbone supplies text conditioning. The readout does not require an additional text-conditioning branch. Architectural details appear in Appendix G.1, and the conditioning ablation is reported in Appendix I.3.

3.4 Learning and Using the Critic

Algorithm 1 summarizes the critic warm-up and subsequent actor–critic training.

To learn these continuation values, we collect trajectories with the rollout policy π_{old} and pair their intermediate states with Monte Carlo return targets. With terminal-only rewards and no discounting, every state in a trajectory has the same observed return R. Different states become distinguishable through regression over many trajectories. The critic minimizes

$$\mathcal{L}_{\text{value}}(\phi) = \mathbb{E}_{\tau \sim \pi_{\text{old}},\, k}\left[\left(V_\phi(s_k) - \widehat{R}_k\right)^2\right], \tag{3}$$

where $\widehat{R}_k$ is the Monte Carlo target with the value-target clipping specified in Appendix G. The reward evaluator supplies supervision only on completed images and need not be differentiable.

Diffusion pretraining provides denoising features but does not calibrate the new readout to reward magnitudes. We therefore begin with a short critic warm-up in which we keep the actor fixed, sample trajectories, and update only the critic using Eq. (3). After warm-up, we update both networks using fresh rollouts, allowing the critic to track the changing policy. The warm-up ablation is given in Appendix I, with training settings in Appendix G.

With these value estimates, we optimize the actor using PPO (Schulman et al., 2017). In our experiments, GAE uses $\gamma = \lambda = 1$, giving the undiscounted Monte Carlo advantage

$$\widehat{A}_k = R - V_\phi(s_k). \tag{4}$$

Let $\widetilde{A}_k$ denote this estimate after the global-batch normalization and clipping described in Appendix G. The actor minimizes

$$\begin{aligned} \mathcal{L}_{\text{policy}}(\theta) &= -\mathbb{E}_{\tau\sim\pi_{\text{old}},\,k}\left[\min\Big(\rho_k(\theta)\widetilde{A}_k, \text{clip}(\rho_k(\theta), 1-\epsilon, 1+\epsilon)\widetilde{A}_k\Big)\right], \\ \rho_k(\theta) &= \frac{\pi_\theta(a_k \mid s_k)}{\pi_{\text{old}}(a_k \mid s_k)}. \end{aligned} \tag{5}$$

Advantages and rollout likelihoods are held fixed during the actor update. The critic is trained with its own regression objective. Thus, the conditional comparison motivated in Sec. 3.2 enters learning through the advantage used to weight each denoising transition.

Algorithm 1 Training diffusion models with a state-dependent critic

Input: pretrained actor θ_0, prompt distribution $\mathcal{D}$, reward r, W critic warm-up iterations, N joint-training iterations, U updates per iteration.

Initialize actor $\theta \leftarrow \theta_0$ and a separate critic ϕ from the pretrained backbone with a value readout.
for $i = 1, \ldots, W + N$ **do**
  Set $\theta_{\text{old}} \leftarrow \theta$ and sample a batch of prompts $y \sim \mathcal{D}$.
  Sample trajectories with π_{old} and record states s_k, actions a_k, and rollout log-probabilities.
  Decode each final latent into x and evaluate its terminal reward $R = r(x, y)$.
  Form Monte Carlo targets $\widehat{R}_k$ from R, with the target clipping in Appendix G.
  if $i > W$ **then**
    Compute $\widehat{A}_k = R - V_\phi(s_k)$ for each recorded state.
    Normalize and clip advantages over the global batch and freeze the resulting $\widetilde{A}_k$.
  end if
  for $u = 1, \ldots, U$ **do**
    Update ϕ by minimizing $\mathcal{L}_{\text{value}}$ (Eq. (3)) on rollout minibatches.
    if $i > W$: update θ with $\mathcal{L}_{\text{policy}}$ (Eq. (5)), using fixed rollout likelihoods and $\widetilde{A}_k$.
  end for
end for

Output: trained actor θ and critic ϕ.
For multiple rewards, sample reward-specific prompts and use the corresponding reward, value head, and configured sample weight.

Multi-reward optimization. The same procedure extends to multiple reward functions $\{r^{(m)}\}_{m=1}^{M}$. We retain one shared critic backbone and attach a separate value head for each objective:

$$V_\phi^{(m)}(s_k) \approx \mathbb{E}_\pi[R^{(m)} \mid s_k], \qquad \widehat{A}_k^{(m)} = R^{(m)} - V_\phi^{(m)}(s_k). \tag{6}$$

Each head learns the scale and state dependence of its own reward, while the expensive latent feature extraction is shared. We train on a mixture of reward-specific prompt sets. Each sample carries its reward type and weight, selects the corresponding value head, and contributes the corresponding regression and policy losses. Dataset proportions and reward weights control the relative emphasis on each objective (Appendix G). This extends the same conditional-value formulation to heterogeneous rewards.

Inference-time steering. The critic also supports inference-time steering because its value prediction is differentiable with respect to the noisy latent, even when its terminal supervision comes from a non-differentiable reward. With actor and critic parameters fixed, we compute

$$g_t = \nabla_{z_t} V_\phi(z_t, t, y) \tag{7}$$

and use this direction to steer sampling toward states with higher predicted continuation values, in the spirit of gradient-based guidance (Dhariwal & Nichol, 2021). For a DDIM-style noise prediction (Song et al., 2021), we use

$$\hat{\epsilon}_\theta(z_t, t, y) = \epsilon_\theta(z_t, t, y) - \eta\sqrt{1-\bar{\alpha}_t}\, g_t, \tag{8}$$

where $\bar{\alpha}_t$ is the cumulative noise-schedule coefficient and $\eta \geq 0$ controls guidance strength. For a flow-based sampler (Liu et al., 2023, Lipman et al., 2023), the corresponding velocity modification is

$$\hat{v}_\theta(z_t, t, y) = v_\theta(z_t, t, y) + \eta_t g_t. \tag{9}$$

Here η_t is a signed coefficient matched to the integration convention: for an Euler step $z_{t+\Delta t} = z_t + \Delta t\, \hat{v}_\theta$, choosing $\Delta t\, \eta_t \geq 0$ makes the added latent displacement follow g_t. This reuses the same continuation-value estimate for test-time guidance. Improvements in realized reward are assessed empirically in Appendix K.2.

4 Experiments

Our experiments address two questions: *(1) Why do we need a critic for optimizing diffusion models?* and *(2) Why use a diffusion model as the critic?* Additional evaluations of multi-reward optimization and inference-time steering are provided in Appendices K.1 and K.2, respectively.

4.1 Why Do We Need a Critic?

Section 3.2 shows that terminal rewards can poorly reflect the expected quality of intermediate states. We first test whether a trained critic ranks these states more accurately, then examine its utility in policy optimization.

State-value ranking. We evaluate the trained diffusion critic on the state-ranking task in Sec. 3.2. For states from the same prompt and denoising stage, we compare rankings induced by critic predictions and final trajectory rewards against the continuation-based state-value reference. Table 1 reports the percentage of state pairs ranked consistently with this reference.

The critic achieves 69.0%, 91.0%, and 94.0% accuracy at the early, middle, and late stages, compared with 58.8%, 87.5%, and 92.7% for final rewards. The gains are 10.2, 3.5, and 1.3 percentage points, respectively. The largest improvement occurs early in denoising, where the final-reward ranking is close to random guessing and uncertainty about the eventual outcome is greatest.

These results support using a learned state value to estimate the expected quality of a partial trajectory. This provides the state-dependent reference used to center terminal returns during actor updates.

Single-reward policy optimization. We then evaluate whether the actor–critic formulation is useful in full policy training.

The single-reward benchmarks cover text-image alignment (CLIP), human preference (HPSv2.1 and PickScore), and verifiable constraints (GenEval and OCR). Table 2 summarizes both backbones.

Our runs use the matched rollout batches specified in Appendix G. Flow-GRPO, DiffusionNFT, and AWM retain their source-reported scores and settings. Their training objectives and evaluation protocols are detailed in Appendix G.3.

Table 2: Benchmark results on SD1.5 and SD3.5M. Evaluation protocol details appear in Appendix G.3.

Method	CLIP	HPSv2.1	PickScore	GenEval	OCR
SD1.5					
Base	0.2250	0.1727	20.53	0.1558	0.066
DDPO	0.3233	0.2752	21.87	0.5455	0.2351
DanceGRPO	0.3342	0.3779	24.06	0.5888	0.2518
Ours	**0.3431**	**0.3912**	**24.58**	**0.6392**	**0.2648**
SD3.5M					
Base	0.2275	0.1750	21.71	0.2013	0.0002
DDPO	0.2822	0.2246	19.76	div	div
DanceGRPO	0.3006	0.3604	23.04	0.9620	0.9278
Flow-GRPO, KL	–	–	23.31	0.95	0.92
DiffusionNFT	0.293	0.331	**23.80**	0.94	0.91
AWM	–	–	23.11	**0.985**	0.92
Ours	**0.3029**	**0.3642**	23.20	0.9758	**0.9375**

Table 3: Training cost on HPS (32 H100s). Extra memory is per GPU.

	FlowGRPO	Ours
Time/iter. (s)	84.5	109.0
Relative	1.00×	1.29×
Extra memory (GB/GPU)	–	~ 3

Across the five single-reward tasks, our method outperforms DDPO and DanceGRPO on both backbones wherever the baselines converge (Table 2). Relative to DanceGRPO, HPSv2.1 increases from 0.3779 to 0.3912 on SD1.5 and from 0.3604 to 0.3642 on SD3.5M. CLIP and PickScore also improve on both backbones.

For verifiable rewards, GenEval increases from 0.5888 to 0.6392 on SD1.5 and from 0.9620 to 0.9758 on SD3.5M. OCR increases from 0.2518 to 0.2648 and from 0.9278 to 0.9375, respectively. DDPO diverges on both verifiable tasks in the reported SD3.5M runs, while DanceGRPO and our method train successfully. Figure 4 provides qualitative examples.



Figure 4: Qualitative comparison. The GRPO label denotes DanceGRPO.

The published SD3.5M results provide additional context. Our GenEval score (0.9758) exceeds the reported Flow-GRPO and DiffusionNFT scores (0.95 and 0.94), but is below AWM (0.985). Our OCR score (0.9375) is higher than the reported 0.91–0.92 range of these methods. These numerical comparisons span different training and evaluation settings. For PickScore, our SD3.5M score is 23.20, compared with 23.31 for Flow-GRPO, 23.80 for DiffusionNFT, and 23.11 for AWM. Evaluation protocol differences are detailed in Appendix G.3.

The policy-training results complement the state-ranking comparison: the critic estimates intermediate state quality more accurately, and the resulting actor–critic framework improves the tested single-reward objectives.

Computational cost. Under the FlowGRPO HPS setting, our method takes 109.0s per iteration versus 84.5s for FlowGRPO, with approximately 3 GB extra memory per GPU (Table 3). This 29% time overhead is an acceptable cost for state-dependent value estimation within the same 32-H100 configuration.

4.2 Why Use Diffusion Models as Critics?

A useful critic must predict the return from an intermediate denoising state and distinguish between alternative continuations. We evaluate these two properties using three trained critics—CLIP, HPS, and our diffusion critic—on a held-out test set. We then examine downstream policy performance and critic runtime.

Value prediction and action ranking. We train each critic to predict returns and measure the mean squared error between its predictions and the reference value targets on the test set. This assesses the accuracy of the learned value estimates, independently of the final performance of an optimized policy.

Table 4: Test-set value prediction and action ranking.

Critic	MSE ↓	Rank acc. (%) ↑
CLIP	6.965×10^{-4}	74.2
HPS	6.603×10^{-4}	64.0
Ours	$\mathbf{5.534 \times 10^{-4}}$	**78.5**

To assess whether these estimates distinguish alternative actions, we sample two actions from the same denoising state and use the critic to rank their resulting states. We then continue both trajectories to completion and compare the predicted ranking with the ordering of their final rewards. Ranking accuracy is the percentage of evaluated pairs for which these orderings agree.

As shown in Table 4, our critic achieves an MSE of 5.534×10^{-4}, reducing the error by 20.5% relative to CLIP and 16.2% relative to HPS. Its ranking accuracy is 78.5%, exceeding CLIP (74.2%) by 4.3 percentage points and HPS (64.0%) by 14.5 percentage points.

The two metrics capture different aspects of critic quality: HPS has lower MSE than CLIP but substantially lower ranking accuracy. Our diffusion critic performs best on both, supporting its ability to estimate returns and distinguish more promising sampled continuations. The ranking comparison uses realized terminal rewards. It measures agreement with observed outcomes rather than exact recovery of expected action values.

Policy performance and evaluation cost. Table 5a compares the final HPSv2.1 reward achieved with different critic backbones, using the implementations described in Appendix G. The diffusion critic reaches 0.3912, compared with 0.3323 for CLIP, 0.3612 for BLIP, and 0.3495 for HPSv2.1. It exceeds the strongest pixel-space alternative, BLIP, by 0.0300.

These downstream rewards complement the test-set value and ranking results: the diffusion critic combines more accurate return prediction and stronger action discrimination with higher reward after policy training.

Table 5: Critic comparison: diffusion features versus CLIP (Radford et al., 2021), BLIP (Li et al., 2022), and HPSv2.1 (Wu et al., 2023c).

(a) Ablation on critic design (HPSv2.1 reward).

Critic	CLIP	BLIP	HPSv2.1	Ours
Results ↑	0.3323	0.3612	0.3495	**0.3912**

(b) Critic runtime (seconds).

Critic	Ours	CLIP	BLIP	HPSv2.1
Value eval.	10	33	35	28
Training	20	78	86	52

The same representation choice also affects cost. In the reported configuration, value computation takes 10s for our critic versus 28–35s for the pixel-space alternatives, while critic training takes 20s versus 52–86s (Table 5b). The alternatives therefore take 2.8–$3.5\times$ as long for value computation and 2.6–$4.3\times$ as long for critic training.

Operating on the latent state avoids intermediate image reconstruction and decoding during critic evaluation. Terminal images are still decoded for reward computation.

An additional ablation on critic-head conditioning is provided in Appendix I.3.

5 Conclusion

We present a practical actor–critic framework for post-training diffusion models, addressing trajectory-level credit assignment through state-dependent value estimation. A pretrained diffusion backbone provides a timestep-aware critic that predicts continuation values directly on noisy latents, while value pretraining supports stable PPO updates. Experiments examine the role of the critic and its diffusion representation, with improved single-reward performance in our controlled comparisons. The same framework extends to multiple reward objectives and reuses the learned critic for inference-time steering. These components establish a practical recipe for value-based diffusion post-training.

AI USE STATEMENT

In this work, we used generative AI tools to assist with designing and providing feedback on research methodology and experiments, implementing methods, translation, qualitative and thematic data analysis, and interpretation of results. We also used these tools to draft parts of the manuscript, transcribe recordings of research material, brainstorm ideas, source and search for information, improve the readability of the manuscript, and suggest its structure. The authors take responsibility for the final content of this work, including all text, claims, analyses, and artifacts produced with the assistance of generative AI.

REFERENCES

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms. *arXiv preprint arXiv:2402.14740*, 2024.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022a.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. Constitutional ai: Harmlessness from ai feedback, 2022b. URL https://arxiv.org/abs/2212.08073.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=YCWjhGrJFD.

Kevin Clark, Paul Vicol, Kevin Swersky, and David J. Fleet. Directly fine-tuning diffusion models on differentiable rewards. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=1vmSEVL19f.

Cheng Cui, Ting Sun, Manhui Lin, Tingquan Gao, Yubo Zhang, Jiaxuan Liu, Xueqing Wang, Zelun Zhang, Changda Zhou, Hongen Liu, Yue Zhang, Wenyu Lv, Kui Huang, Yichao Zhang, Jing Zhang, Jun Zhang, Yi Liu, Dianhai Yu, and Yanjun Ma. Paddleocr 3.0 technical report, 2025. URL https://arxiv.org/abs/2507.05595.

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.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Zheng Ding and Weirui Ye. TreeGRPO: Tree-advantage GRPO for online RL post-training of diffusion models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=3rZdp4TmUb.

Arnaud Doucet et al. *Sequential Monte Carlo methods in practice*, volume 1. Springer, 2001.

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.

Jiajun Fan, Shuaike Shen, Chaoran Cheng, Yuxin Chen, Chumeng Liang, and Ge Liu. Online reward-weighted fine-tuning of flow matching with wasserstein regularization. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=2IoFFexvuw.

Tiantian Fan, Lingjun Liu, Yu Yue, Jiaze Chen, Chengyi Wang, Qiying Yu, Chi Zhang, Zhiqi Lin, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Bole Ma, Mofan Zhang, Gaohong Liu, Ru Zhang, Haotian Zhou, Cong Xie, Ruidong Zhu, Zhi Zhang, Xin Liu, Mingxuan Wang, Lin Yan, and Yonghui Wu. Truncated proximal policy optimization, 2025b. URL https://arxiv.org/abs/2506.15050.

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.

Hiroki Furuta, Heiga Zen, Dale Schuurmans, Aleksandra Faust, Yutaka Matsuo, Percy Liang, and Sherry Yang. Improving dynamic object interactions in text-to-video generation with ai feedback. *arXiv preprint arXiv:2412.02617*, 2024.

Ruiqi Gao, Emiel Hoogeboom, Jonathan Heek, Valentin De Bortoli, Kevin P. Murphy, and Tim Salimans. Diffusion meets flow matching: Two sides of the same coin, 2024. URL https://diffusionflow.github.io/.

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.

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

Dong Guo, Faming Wu, Feida Zhu, Fuxing Leng, Guang Shi, Haobin Chen, Haoqi Fan, Jian Wang, Jianyu Jiang, Jiawei Wang, et al. Seed1. 5-vl technical report. *arXiv preprint arXiv:2505.07062*, 2025b.

Shashank Gupta, Chaitanya Ahuja, Tsung-Yu Lin, Sreya Dutta Roy, Harrie Oosterhuis, Maarten de Rijke, and Satya Narayan Shukla. A simple and effective reinforcement learning method for text-to-image diffusion fine-tuning. *arXiv preprint arXiv:2503.00897*, 2025.

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

Jian Hu, Jason Klein Liu, Haotian Xu, and Wei Shen. Reinforce++: An efficient rlhf algorithm with robustness to both prompt and reward models. *arXiv preprint arXiv:2501.03262*, 2025.

Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. *Advances in Neural Information Processing Systems*, 36:78723–78747, 2023.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Sham M Kakade. A natural policy gradient. *Advances in neural information processing systems*, 14, 2001.

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.

Vijay R Konda and John N Tsitsiklis. Onactor-critic algorithms. *SIAM journal on Control and Optimization*, 42(4):1143–1166, 2003.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 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.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pp. 12888–12900. PMLR, 2022.

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, 2025a. URL https://arxiv.org/abs/2507.21802.

Shufan Li, Konstantinos Kallidromitis, Akash Gokul, Yusuke Kato, and Kazuki Kozuka. Aligning diffusion models by optimizing human utility. *Advances in Neural Information Processing Systems*, 37:24897–24925, 2024.

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

Ziniu Li, Tian Xu, Yushun Zhang, Zhihang Lin, Yang Yu, Ruoyu Sun, and Zhi-Quan Luo. Remax: A simple, effective, and efficient reinforcement learning method for aligning large language models. *arXiv preprint arXiv:2310.10505*, 2023.

Zhanhao Liang, Yuhui Yuan, Shuyang Gu, Bohan Chen, Tiankai Hang, Mingxi Cheng, Ji Li, and Liang Zheng. Aesthetic post-training diffusion models from generic preferences with step-by-step preference optimization. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 13199–13208, 2025.

Zhengyang Liang, Hao He, Ceyuan Yang, and Bo Dai. Scaling laws for diffusion transformers. *arXiv preprint arXiv:2410.08184*, 2024.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Shanchuan Lin, Xin Xia, Yuxi Ren, Ceyuan Yang, Xuefeng Xiao, and Lu Jiang. Diffusion adversarial post-training for one-step video generation. *arXiv preprint arXiv:2501.08316*, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky T. Q. Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code, 2024. URL https://arxiv.org/abs/2412.06264.

Jiashun Liu, Johan Obando-Ceron, Han Lu, Yancheng He, Weixun Wang, Wenbo Su, Bo Zheng, Pablo Samuel Castro, Aaron Courville, and Ling Pan. Asymmetric proximal policy optimization: mini-critics boost llm reasoning. *arXiv preprint arXiv:2510.01656*, 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. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b. URL https://openreview.net/forum?id=oCBKGw5HNf.

Runtao Liu, Haoyu Wu, Ziqiang Zheng, Chen Wei, Yingqing He, Renjie Pi, and Qifeng Chen. Videodpo: Omni-preference alignment for video diffusion generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 8009–8019, 2025c.

Xingchao Liu, Chengyue Gong, and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

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, 2025.

David McAllister, Songwei Ge, Brent Yi, Chung Min Kim, Ethan Weber, Hongsuk Choi, Haiwen Feng, and Angjoo Kanazawa. Flow matching policy gradients, 2025. URL https://arxiv.org/abs/2507.21053.

Zichen Miao, Jiang Wang, Ze Wang, Zhengyuan Yang, Lijuan Wang, Qiang Qiu, and Zicheng Liu. Training diffusion models towards diverse image generation with reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10844–10853, 2024.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pp. 1928–1937. PmLR, 2016.

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.

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.

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

Mihir Prabhudesai, Russell Mendonca, Zheyang Qin, Katerina Fragkiadaki, and Deepak Pathak. Video diffusion alignment via reward gradients, 2024. URL https://arxiv.org/abs/2407.08737.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

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.

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.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

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

Team Seawead, Ceyuan Yang, Zhijie Lin, Yang Zhao, Shanchuan Lin, Zhibei Ma, Haoyuan Guo, Hao Chen, Lu Qi, Sen Wang, et al. Seaweed-7b: Cost-effective training of video generation foundation model. *arXiv preprint arXiv:2504.08685*, 2025.

ByteDance Seed, Jiaze Chen, Tiantian Fan, Xin Liu, Lingjun Liu, Zhiqi Lin, Mingxuan Wang, Chengyi Wang, Xiangpeng Wei, Wenyuan Xu, et al. Seed1. 5-thinking: Advancing superb reasoning models with reinforcement learning. *arXiv preprint arXiv:2504.13914*, 2025.

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.

Dingyuan Shi, Yong Wang, Hangyu Li, and Xiangxiang Chu. Preference alignment for diffusion model via explicit denoised distribution estimation. *arXiv preprint arXiv:2411.14871*, 2024.

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In Eric P. Xing and Tony Jebara (eds.), *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 387–395, Bejing, China, 22–24 Jun 2014. PMLR. URL https://proceedings.mlr.press/v32/silver14.html.

Raghav Singhal, Zachary Horvitz, Ryan Teehan, Mengye Ren, Zhou Yu, Kathleen McKeown, and Rajesh Ranganath. A general framework for inference-time scaling and steering of diffusion models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=Jp988ELppQ.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=St1giarCHLP.

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. *arXiv preprint arXiv:2011.13456*, 2020.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021, 2020.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.

Yibin Wang, Yuhang Zang, Hao Li, Cheng Jin, and Jiaqi Wang. Unified reward model for multimodal understanding and generation. *arXiv preprint arXiv:2503.05236*, 2025.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

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*, 2025a.

Luhuan Wu, Brian L Trippe, Christian Naesseth, David Blei, and John P Cunningham. Practical and asymptotically exact conditional sampling in diffusion models. *Advances in Neural Information Processing Systems*, 2023a.

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*, 2023b.

Xiaoshi Wu, Keqiang Sun, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score: Better aligning text-to-image models with human preference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2096–2105, 2023c.

Ziyi Wu, Anil Kag, Ivan Skorokhodov, Willi Menapace, Ashkan Mirzaei, Igor Gilitschenski, Sergey Tulyakov, and Aliaksandr Siarohin. DenseDPO: Fine-grained temporal preference optimization for video diffusion models. *NeurIPS*, 2025b.

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.

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, 2024a.

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

Yuanyang Yin, Yaqi Zhao, Mingwu Zheng, Ke Lin, Jiarong Ou, Rui Chen, Victor Shea-Jay Huang, Jiahao Wang, Xin Tao, Pengfei Wan, et al. Towards precise scaling laws for video diffusion transformers. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 18155–18165, 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.

Chuheng Zhang, Wei Shen, Li Zhao, Xuyun Zhang, Xiaolong Xu, Wanchun Dou, and Jiang Bian. Policy filtration for RLHF to mitigate noise in reward models. In *Forty-second International Conference on Machine Learning*, 2025a. URL https://openreview.net/forum?id=L8hYdTQVcs.

Jiacheng Zhang, Jie Wu, Weifeng Chen, Yatai Ji, Xuefeng Xiao, Weilin Huang, and Kai Han. Onlinevpo: Align video diffusion model with online video-centric preference optimization. *arXiv preprint arXiv:2412.15159*, 2024a.

Tao Zhang, Cheng Da, Kun Ding, Huan Yang, kun jin, Yan Li, Tingting Gao, Di ZHANG, Shiming Xiang, and Chunhong Pan. Diffusion model as a noise-aware latent reward model for step-level preference optimization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b. URL https://openreview.net/forum?id=YB9VGCClv9.

Xiangcheng Zhang, Haowei Lin, Haotian Ye, James Zou, Jianzhu Ma, Yitao Liang, and Yilun Du. Inference-time scaling of diffusion models through classical search. *arXiv preprint arXiv:2505.23614*, 2025c.

Yinan Zhang, Eric Tzeng, Yilun Du, and Dmitry Kislyuk. Large-scale reinforcement learning for diffusion models. In *European Conference on Computer Vision*, pp. 1–17. Springer, 2024b.

Hanyang Zhao, Haoxian Chen, Ji Zhang, David Yao, and Wenpin Tang. Score as action: Fine tuning diffusion generative models by continuous-time reinforcement learning. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=V5HEX6stS2.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 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, 2025. URL https://arxiv.org/abs/2510.01982.

Dingwei Zhu, Shihan Dou, Zhiheng Xi, Senjie Jin, Guoqiang Zhang, Jiazheng Zhang, Junjie Ye, Mingxu Chai, Enyu Zhou, Ming Zhang, Caishuang Huang, Yunke Zhang, Yuran Wang, and Tao Gui. Vrpo: Rethinking value modeling for robust rl training under noisy supervision, 2025. URL https://arxiv.org/abs/2508.03058.

A Appendix Overview

This appendix is organized as follows:

- Section B discusses the limitations of our approach.
- Section C reports the licenses for existing assets.
- Section D summarizes the notation used throughout the paper.
- Section E reviews additional related work.
- Section F reports further results on inference-time steering.
- Section G provides additional implementation and methodological details.
- Section H shows additional visual results and comparisons.
- Section I reports additional ablations on training choices.
- Section J discusses multi-reward optimization and reward hacking.
- Section K evaluates multi-reward optimization and inference-time steering.

B Limitations

Our approach still has several limitations. First, we focus exclusively on image diffusion models. Extending our actor–critic framework to video or other modalities remains an open challenge. Second, although our multi–value-head design supports multi-reward optimization, the optimization efficiency for each individual reward can degrade compared to single-reward training. Third, our method relies on time discretization when sampling from the SDE, which may introduce discretization error and limit the RL efficiency. We leave these directions for future work.

C Licenses for existing assets

SD3.5: MIT License (https://github.com/Stability-AI/sd3.5)

HPSv2: Apache-2.0 License (https://github.com/tgxs002/HPSv2)

CLIP: MIT License (https://github.com/openai/CLIP)

PickScore: MIT License (https://github.com/yuvalkirstain/pickscore)

GenEval: MIT License (https://github.com/djghosh13/geneval)

PaddleOCR: Apache-2.0 License (https://github.com/PADDLEPADDLE/PADDLEOCR)

BLIP: BSD-3-Clause license (https://github.com/salesforce/BLIP?tab=BSD-3-Clause-1-ov-file)

D Notation

Table 6 summarizes the notation and abbreviations used in this paper.

E Extended Related Work

Credit assignment in diffusion RL. A central challenge in diffusion RL is credit assignment along the denoising trajectory: rewards are typically observed only on the final decoded image, while optimization must assign credit to intermediate denoising steps. Existing methods address this problem in different ways. Early RL-based approaches such as DDPO and DPOK mainly optimize terminal rewards with policy gradients (Black et al., 2024, Fan et al., 2023). Preference-based methods have explored denser or step-aware supervision, for example through temporal discounting (Yang et al., 2024b), step-aware preference optimization (Liang et al., 2025), and explicit denoised-distribution estimation that links intermediate steps to the final denoised outcome (Shi et al., 2024). Within the GRPO family, current work avoids explicit value learning and instead relies on

Symbol	Meaning	Symbol	Meaning
x_0	Clean data sample (image latent).	s_k	State at step k, e.g. (x_{t_k}, t_k, y).
x_t	Noisy latent at time t.	a_k	Action at step k (predicted denoising direction).
x_1	Standard normal latent in OT-path flow matching.	r_k	Reward at step k (zero for $k < T$, terminal-only at $k = T$).
t, t_k	Continuous time or discrete timestep.	R_k	Return-to-go $\sum_{l=0}^{T-k} \gamma^l r_{k+l}$.
T	Number of reverse-diffusion / flow steps.	A_k	Advantage estimate at step k.
y	Text prompt / conditioning input.	$V_\phi(s_k)$	Value function (critic) at state s_k.
$v_\theta(x_t, t)$	Policy / flow network prediction (velocity / noise).	δ_k	TD residual $r_k + \gamma V_\phi(s_{k+1}) - V_\phi(s_k)$.
$\hat{x}_0, \hat{x}_1$	Reconstructions used in the stochastic flow sampler.	γ	Discount factor in RL.
Δt	Step size between adjacent time steps.	λ	GAE parameter (bias–variance trade-off).
ϵ	Standard Gaussian noise, $\epsilon \sim \mathcal{N}(0, I)$.	$\pi_\theta(a_k \mid s_k, t_k)$	Stochastic policy at step k.
σ_t	Noise scale / stochasticity at time t.	π_{ref}	Frozen pretrained reference policy.
$\alpha_t, \bar{\alpha}_t$	DDPM schedule, $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$.	$r_k(\theta)$	Policy likelihood ratio $\frac{\pi_\theta(a_k \mid s_k, t_k)}{\pi_{\theta_{\text{old}}}(a_k \mid s_k, t_k)}$ in PPO.
$v(x_t, t)$	Ground-truth OT-path velocity $x_1 - x_0$.	$J(\theta)$	Expected return objective.
$\mathcal{T}_t$	One-step sampler transition (DDIM / Euler).	τ	Trajectory $\{x_{t_k}\}_{k=0}^{T}$.
K	Number of particles in SMC.	$x_t^{(i)}$	i-th particle in SMC at time t.
$w_t^{(i)}$	Normalized weight of particle i in SMC.	g_t	Critic gradient $\nabla_{x_t} V_\phi(x_t, t, y)$ for guidance.
η	Guidance scale for critic-based steering.		
PPO	Proximal Policy Optimization.	**GAE**	Generalized Advantage Estimation.
MDP	Markov Decision Process.	**BoN**	Best-of-N sampling baseline.
SMC	Sequential Monte Carlo sampling.	**UNet**	U-shaped convolutional backbone.
DiT	Diffusion Transformer backbone.	**VAE**	Variational Autoencoder.
HPS	Human Preference Score (HPSv2.1).	**OCR**	Optical Character Recognition reward.
SD1.5	Stable Diffusion 1.5 backbone.	**SD3.5M**	Stable Diffusion 3.5M backbone.

Table 6: **Summary of the notation and abbreviations used in this paper.**

relative group signals for trajectory optimization (Liu et al., 2025b, Xue et al., 2025b). More recent follow-up work pushes this direction further with finer-grained reward assignment, including granular reward assessment in G^{2}RPO, dense step-wise rewards in DenseGRPO, and tree-structured temporal credit assignment in TreeGRPO (Zhou et al., 2025, Deng et al., 2026, Ding & Ye, 2026). Our work is complementary to these approaches: instead of redesigning relative or dense reward signals, we learn an explicit critic for trajectory-level actor-critic optimization.

Representation and state-space alignment. Another underexplored issue is the representation space in which alignment signals are defined. Many existing diffusion alignment pipelines evaluate decoded images with external reward models or vision-language encoders even when optimization concerns intermediate denoising states (Zhao et al., 2025, Black et al., 2024). This creates a mismatch between the states visited by the policy and the states evaluated by the reward model or critic. Several recent works begin to address this issue more directly. SPO introduces step-aware preference modeling at each denoising step (Liang et al., 2025). DDE explicitly connects intermediate states to the terminal denoised distribution for preference alignment (Shi et al., 2024), and LPO argues that diffusion backbones are naturally suited for modeling rewards directly on noisy latent states (Zhang et al., 2025b). These works move toward diffusion-native representations, but they do not learn an explicit value critic for online actor-critic optimization. Our work shares the same intuition that alignment signals should be defined in the noisy latent space, but uses this representation to learn a timestep-conditioned value function so that actor and critic operate in the same state space.

Diffusion/Flow Models. Diffusion-based generative models have achieved state-of-the-art performance in visual synthesis. Denoising diffusion probabilistic models (DDPMs) (Ho et al., 2020) formulate generation as reversing a fixed Markovian noising process and optimize a variational bound that can be interpreted as denoising score matching (Song & Ermon, 2019). Score-based diffusion models further generalize this view by describing the forward and reverse processes as stochastic differential equations and learning time-dependent score functions, which yields a unified framework for diffusion and Langevin-based samplers and enables powerful conditional generation for diverse vision tasks (Song et al., 2020). Flow-matching models directly learning continuous-time vector fields that transport a simple prior to the data distribution within the framework of continuous normalizing flows (Lipman et al., 2023, Liu et al., 2023). Recent analyses show that Gaussian flow matching and diffusion models are closely related and can be viewed as two parameterizations of the same underlying continuous-time generative process, offering complementary design choices in terms of objective, architecture, and discretization (Lipman et al., 2024, Gao et al., 2024). In practice, flow-matching models have emerged as an attractive alternative for building fast, high-fidelity generative priors for modern vision applications.

Post-training for Foundation Models. Foundation models are typically obtained by large-scale pre-training on generic corpora, and then adapted to downstream use via a post-training pipeline. Modern post-training stacks begin with supervised instruction tuning on curated demonstrations to teach the model task formats and high-level alignment objectives (Ouyang et al., 2022). On top of this, reinforcement learning from human feedback (RLHF) trains a reward model from pairwise preferences and then optimizes the policy with reinforcement learning (e.g., PPO), improving helpfulness, safety, and calibration beyond pure supervised learning (Bai et al., 2022a, Lambert et al., 2024). To reduce dependence on expensive human labels, recent work explores reinforcement learning from AI feedback (Bai et al., 2022b) (RLAIF) and reinforcement learning with verifiable rewards (RLVR) (Shao et al., 2024, Team et al., 2025), where the reward signal is derived from automatically checkable criteria such as unit tests, program analyzers, or self-consistency checks, making it particularly suitable for code generation and long-horizon reasoning tasks. In parallel, direct preference optimization objectives such as DPO and related methods bypass explicit reward modeling and directly fit the policy to preference data while regularizing towards the pre-trained model (Rafailov et al., 2023). Recent reasoning-centric foundation models, including the OpenAI o1 (Jaech et al., 2024) series and DeepSeek-R1 (Guo et al., 2025a), instantiate multi-stage post-training pipelines that combine large-scale instruction tuning, RLHF-/RLVR-style optimization, and process-level supervision over chain-of-thought traces, yielding substantial gains on math, coding, and scientific reasoning benchmarks. Overall, instruction tuning, RLHF/RLAIF, RLVR-style verifiable-reward training, and direct preference optimization have emerged as the main families of post-training methods for aligning general-purpose foundation models with target tasks and user preferences.

RL with Critics for LLM Post-training A large fraction of RL post-training for language models follows a critic-based design, where a value function is learned jointly with the policy to handle sparse, sequence-level feedback. In RLHF pipelines such as InstructGPT (Ouyang et al., 2022) and Constitutional AI (Bai et al., 2022b), a reward model is first trained from human or AI preferences, and PPO with a value head on the LM is then used to maximize the reward under KL regularization (Zhu et al., 2025). Similar architectures appear in RL with verifiable or tool-based rewards, where the critic must propagate terminal signals back through long action sequences (Liu et al., 2025a, Fan et al., 2025b). Recent work shows that careful choices of value normalization, credit assignment, and trajectory filtering allow PPO-style critics to scale to harder reasoning and long chain-of-thought settings (Yuan et al., 2024, Seed et al., 2025, Guo et al., 2025b, Zhang et al., 2025a). These results suggest that strong, well-trained critics remain central when we need high performance on complex generation tasks. Our work follows this line but adapts the value architecture and training to the latent, multi-step structure of diffusion models.

Policy Gradient Methods. Policy-gradient methods optimize a parameterized policy by directly ascending the gradient of the expected return, rather than relying on value-based bootstrapping. Early work such as REINFORCE (Williams, 1992) established the likelihood-ratio estimator and showed that subtracting a learned baseline can substantially reduce gradient variance. Actor-critic architectures extend this idea by coupling a policy (actor) with a learned value function (critic), enabling more sample-efficient and stable learning in both continuous and discrete control (Konda & Tsitsiklis, 2003, Mnih et al., 2016). Building on the natural policy gradient (Kakade, 2001), trust-region methods such as TRPO (Schulman et al., 2015) constrain policy updates within a KL-

divergence trust region to prevent catastrophic performance collapse, while generalized advantage estimation further improves the bias–variance trade-off of the policy-gradient estimator (Schulman et al., 2016). Proximal Policy Optimization (PPO) simplifies TRPO by replacing hard trust-region constraints with clipped surrogate objectives, yielding a robust and computationally efficient on-policy algorithm that has become the de facto choice in many large-scale RL applications (Schulman et al., 2017). In parallel, deterministic policy-gradient methods and their deep variants (e.g., DDPG) extend policy-gradient ideas to off-policy learning in continuous action spaces via critic-based Q-function estimation and target networks (Silver et al., 2014, Lillicrap et al., 2015). These developments form the algorithmic backbone for modern reinforcement-learning-based post-training of foundation models, where PPO-style policy gradients, variance-reduced advantage estimation, and critic-based shaping are widely used to optimize reward models derived from human or verifiable feedback.

F More Results on Inference Time Steering

In this section, we provide more inference time steering results on CLIP and PickScore. The results are shown in Fig. 5. We can observe that critic guidance can consistently improve on baseline and BoN.

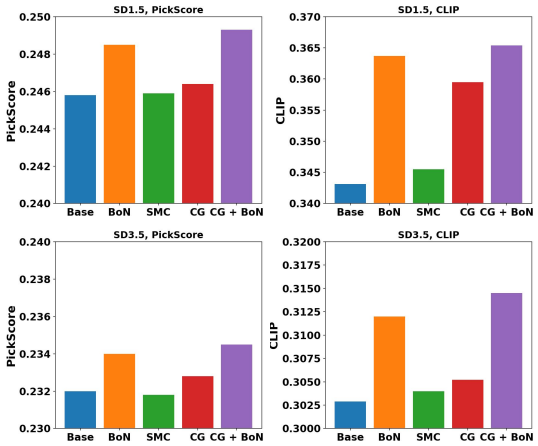

Figure 5: Inference steering results on CLIP and PickScore.

G Details

G.1 Training Details

Our controlled policy-training benchmarks use SD1.5 and SD3.5M with CLIP, HPSv2.1, PickScore, GenEval, and OCR rewards. We compare DDPO (Black et al., 2024), DanceGRPO (Xue et al., 2025b), and our method over 300 iterations. The methods generate the same number of images per iteration: DanceGRPO samples groups per prompt, whereas DDPO and our method use one image per prompt and more distinct prompts. These are matched rollout budgets, not matched total compute. For multi-reward optimization, we use equal weights 1:1:1 for CLIP, HPSv2.1, and GenEval, and mix their prompt sets in the same 1:1:1 ratio. Fixed-policy analyses use SD3.5M with separate fitting budgets and no actor updates.

Optimization. We use AdamW (Loshchilov & Hutter, 2019) to optimize both the policy and value networks in all experiments. The momentum coefficients are set to $(\beta_1, \beta_2) = (0.9, 0.95)$ and the weight decay is 1×10^{-4}. The policy learning rate is fixed to 1×10^{-4} across all settings for a fair comparison. Each run is trained for 300 iterations, and every iteration consists of 4 gradient update steps.

SD1.5. For SD1.5 experiments, we draw a large batch of 3,840 images per iteration. Given 4 update steps, the effective batch size per gradient update is $3840/4 = 960$ samples. The PPO ratio clipping

range is set to 1×10^{-5}. We clip the advantages to the range $[-10, 10]$ and the value targets to $[-5, 5]$. The learning rate of the value network is 1×10^{-4}. Experiments on SD1.5 require 8 H100 GPUs.

SD3.5M. For SD3.5M experiments, we sample 384 images per iteration for the human-preference reward and the CLIP-based reward. For other non-differentiable rewards, we use a larger batch of 3,840 samples. Empirically, we find that small batch sizes lead to unstable training for DDPO/GRPO-style updates, hence the larger batch is beneficial in this case. The ratio clipping range is set to 1×10^{-4}. We keep the same advantage clipping range $[-10, 10]$ and value clipping range $[-5, 5]$. The value network learning rate is set to 1×10^{-3}. Experiments on SD1.5 require 32 H100 GPUs.

Sampling. During training, the sampling configuration is aligned with that used at inference time. For SD1.5, we use the DDIM (Song et al., 2021) sampler with 50 diffusion steps. For SD3.5M, we use the Discrete Euler sampler with 40 steps. Unless otherwise specified, classifier-free guidance is disabled ($\mathrm{CFG} = 1.0$) in all experiments.

Normalization. After computing the per-sample advantages, we apply advantage normalization over the global batch. Concretely, we compute the mean and standard deviation of all advantages in the batch and normalize as

$$A_k = \frac{A_k - A_{\text{mean}}}{A_{\text{std}}}. \tag{10}$$

We do not apply any additional normalization to rewards or value targets.

Inference Details At inference time, we use the same sampling configurations as in the training phase. For SD1.5 models, we employ DDIM with 50 sampling steps. For SD3.5M models, we use the Discrete Euler sampler with 40 steps. Classifier-free guidance is disabled for all reported results.

Multiple Reward Experiment Details To support multiple reward signals, we implement a unified dataset that manages several prompt sets corresponding to different rewards. At each training iteration, we sample prompts from these datasets according to the proportions specified in the configuration. Each returned data item contains the following fields: reward type, prompt, weight, and metadata. The metadata field is empty for human-preference rewards. For GenEval, it encodes the required number of objects or the target color. For OCR text-rendering rewards, it stores the desired characters that should appear in the generated image. By assigning different weights to different reward types, we can balance their relative contributions to the overall learning signal.

Critic Attention Layers We augment the diffusion transformer with critic attention layers inserted at layers 7, 15, and 23. Each critic attention layer contains a single learnable query token that aggregates information from the intermediate visual tokens. After passing through all critic attention layers, we concatenate the resulting query tokens and project them to a scalar prediction, which serves as the value estimate.

Within each critic attention layer, the visual tokens are first processed by an RMSNorm. We then obtain the query, key, and value projections (Q/K/V). Before feeding them into the attention module, we apply QKNorm to Q and K to improve numerical stability. The attention module produces a single output token corresponding to the learned query. This token is further processed by a feed-forward network (FFN) to refine the value representation.

Time Conditioning We empirically find that incorporating the diffusion timestep is crucial for the critic. We therefore adopt an adaptive layer-normalization (AdaLN) style time conditioning in the critic attention layers. Specifically, the learned diffusion time embedding is first processed by an MLP to produce per-channel *scale*, *shift*, and *gate* coefficients. The learnable query token is scaled and shifted before the RMSNorm using the corresponding coefficients. After the attention operation, the output token is gated by the residual input using the gate coefficient. The MLP also produces scale, shift, and gate coefficients for modulating the FFN. The overall architecture of the critic attention layer and its time conditioning is illustrated in Fig. 6.

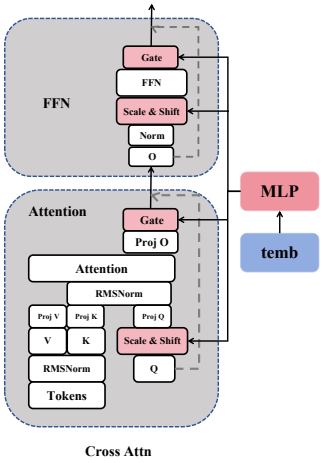

Figure 6: Architecture of the critic attention layer.

G.2 GAE

We use generalized advantage estimation (GAE) to compute the advantages for policy updates. Given a trajectory $\{(s_t, a_t, r_t)\}_{t=0}^{T}$, we first compute the temporal-difference (TD) residuals

$$\delta_t = r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t), \tag{11}$$

where V_ϕ denotes the value network, and we set $V_\phi(s_{T+1}) = 0$ for terminal states. The GAE advantage at time step t is then defined as

$$A_t^{\mathrm{GAE}} = \sum_{l=0}^{T-t-1} (\gamma\lambda)^l \, \delta_{t+l}. \tag{12}$$

GAE involves two hyperparameters: the discount factor γ and the decay factor λ. The discount factor $\gamma \in [0, 1]$ controls how strongly future rewards are taken into account. Smaller values emphasize immediate rewards. The decay factor $\lambda \in [0, 1]$ governs the bias–variance trade-off of the estimator: $\lambda = 0$ corresponds to a TD(0)-style estimate with higher bias but lower variance, whereas $\lambda = 1$ recovers the Monte-Carlo return with lower bias but higher variance.

In our setting, trajectories are short, and the reward is provided at the end of the trajectory. We therefore set both hyperparameters to $\gamma = 1$ and $\lambda = 1$ in all experiments, which reduces GAE to an undiscounted Monte-Carlo estimator of the advantages. After computing A_t^{GAE}, we apply the global advantage normalization described above.

G.3 Provenance of Published Baseline Results

The Flow-GRPO, DiffusionNFT, and AWM entries in Table 2 are source-reported results rather than reproductions under our 300-iteration protocol. We checked Flow-GRPO v5 (arXiv:2505.05470), DiffusionNFT v2 (2509.16117), and AWM v2 (2509.25050). Missing matching entries are dashes. Untreated base-model results are not substituted for post-training results.

Sources are Flow-GRPO (Liu et al., 2025b), Table 2 (task-metric columns), DiffusionNFT, Table 1 [citation pending], and AWM (Xue et al., 2025a), Tables 1–2. No matching SD1.5 post-training results were verified for these three methods.

Flow-GRPO entries use the KL-regularized task-metric columns of its Table 2, with separate GenEval, OCR, and PickScore models. Its DrawBench PickScore column is a different evaluation and is not substituted. DiffusionNFT's Table 1 reports a single CFG-free model trained with five rewards. Its scores are evaluated on GenEval, OCR, and DrawBench, with a 40-step ODE sampler. The 0.98 GenEval headline from its single-reward learning curve is not mixed into the multi-reward row.

AWM uses GenEval from Table 1 and OCR/PickScore from the standard-training row in Table 2, each from task-specific models. For consistent presentation, the non-base PickScore values from

our controlled runs are multiplied by 100 relative to the original reporting scale. Published baseline scores are retained as reported. Identical reward names can also mask differences in prompts, score transformations, or evaluators. The source-reported entries should therefore not be interpreted as a matched-protocol ranking.

G.4 Reward Models

In our experiments, we employ five reward models: CLIP, HPSv2.1, PickScore, GenEval, and OCR. All of them take the generated image, the associated prompt, and additional metadata as input and output a scalar score, which we directly use as the reward without extra normalization.

CLIP (Radford et al., 2021). CLIP measures the text-image alignment between a generated image and its corresponding prompt. Concretely, we encode the prompt and the image with a pretrained CLIP text encoder and image encoder, and compute the cosine similarity between the resulting embeddings. This similarity serves as the CLIP reward, encouraging the model to generate images that are semantically aligned with the input text. We use ViT-L-14 variant in our experiments.

HPSv2.1 (Wu et al., 2023b). HPSv2.1 is a human-preference score model trained on large-scale human annotations. Given a prompt and an image, it predicts how likely a human would prefer this image among candidates. We use the official pretrained HPSv2.1 checkpoint and feed it our generated images and prompts. The predicted score is used as a reward that reflects human aesthetic and semantic preferences.

PickScore (Kirstain et al., 2023). PickScore is another human-preference model built on top of a CLIP backbone and trained on a large-scale dataset. For each image-prompt pair, PickScore outputs a scalar preference score that correlates with human judgments of overall quality.

GenEval (Ghosh et al., 2023). GenEval is a compositional evaluation benchmark designed to test whether a generated image satisfies fine-grained constraints such as the presence, count, and color of objects described in the prompt. During training, we follow its official evaluation pipeline: for each prompt, the metadata specifies the required object categories, counts, or colors. We then apply the GenEval detectors to the generated image and check whether all constraints are satisfied. The GenEval reward is set to 1 if the image passes all checks and 0 otherwise, encouraging the policy to generate images that strictly follow the compositional instructions.

OCR (Cui et al., 2025). To assess text rendering quality, we use an OCR-based reward. For prompts that require specific characters or strings to appear in the image, the metadata stores the target text. We run an off-the-shelf OCR recognizer on the generated image to extract all visible text and compute the fraction of target characters that are correctly recognized in the OCR output. This fraction is used as the OCR reward, which directly encourages the model to produce images with legible and accurate text. We use paddleocr as the reward model.

G.5 Critic Models

We explore several off-the-shelf vision-language models as the backbone of our critic. In all cases, the critic takes the generated image and its corresponding prompt as input and outputs a scalar score, which we interpret as the value of this image-prompt pair. Concretely, we consider three families of critics: (i) a CLIP-based critic, (ii) a BLIP-based critic, and (iii) a critic initialized from a pretrained reward model. Our ablations are mainly conducted on HPSv2.1, therefore we adopt the HPSv2.1-based critic as the default choice in our experiments.

CLIP Critic. For the CLIP critic, we use a pretrained CLIP model and keep both the image and text encoders frozen. Given a prompt and a generated image, we obtain the text embedding and image embedding from CLIP and compute their cosine similarity. . This design leverages the strong text-image alignment capability of CLIP while keeping the critic architecture simple and stable.

BLIP (Li et al., 2022) Critic. BLIP is a vision-language model originally designed for image captioning and vision-language understanding. Similar to the CLIP critic, we use the BLIP image

encoder and text encoder to extract image and text embeddings for each image-prompt pair. We then compute the cosine similarity between these two embeddings and take it as the BLIP critic score. Compared to CLIP, BLIP provides a different inductive bias focused on captioning and language understanding, offering a complementary critic signal. In our experiments, we use blip-itm-base-coco to initialize the critic model.

Reward-model-initialized Critic. Finally, we investigate using a pretrained reward model as the initialization for our critic. In this setting, we start from the HPSv2.1 checkpoint, which is trained on large-scale human preference data. We use the output reward score directly as the critic value. During our RL training, this HPSv2.1-based critic can be further fine-tuned together with the policy, allowing the critic to stay aligned with human preference while adapting to the distribution induced by our policy.

G.6 Inference Guidance

We compare our method against two inference-time baselines that only use the frozen generator and a pretrained reward/critic model at test time: Best-of-N (BoN) and Sequential Monte Carlo (SMC). Both Baselines operate on top of the same diffusion sampler and use the same reward model or critic as our method to score candidate images.

Best-of-N (BoN). Best-of-N is a widely-used test-time selection baseline. For each input prompt, we independently draw N samples from the diffusion model using the standard sampling configuration described above. We then evaluate each generated image with the corresponding reward model (e.g., CLIP, HPSv2.1, or other task-specific reward) and obtain a scalar score. The final output for this prompt is chosen as the image with the highest reward among the N candidates.

Sequential Monte Carlo (SMC). Sequential Monte Carlo (Doucet et al., 2001) provides a more structured test-time procedure that uses a population of samples (particles) and iteratively refines them guided by the reward/critic. For each prompt p, we maintain K particles $\{x_t^{(i)}\}_{i=1}^K$ along the diffusion time steps $t = 0, \dots, T$. At every step, SMC performs propagation, reweighting, and resampling so that computation is focused on high-reward regions of the sample space.

We initialize particles from the standard Gaussian prior with uniform weights:

$$x_T^{(i)} \sim \mathcal{N}(0, I), \quad w_T^{(i)} = \frac{1}{K}, \qquad i = 1, \dots, K. \tag{13}$$

Then, for each diffusion step $t = T-1, \dots, 0$, we proceed as follows:

Propagation. Each particle is propagated by one step of the diffusion sampler $\mathcal{T}_t$ (e.g., DDIM / Euler update) conditioned on the prompt p:

$$\tilde{x}_{t-1}^{(i)} = \mathcal{T}_t\big(x_t^{(i)}, \epsilon_t^{(i)}\big), \tag{14}$$

where $\epsilon_t^{(i)}$ is the sampler noise and $\tilde{x}_{t-1}^{(i)}$ denotes the propagated (pre-resampling) particle.

Reweighting. We evaluate the critic (or reward model) f_ψ on each propagated particle and convert the scalar score into an unnormalized weight, for example, using an exponential mapping with temperature τ:

$$\tilde{w}_{t-1}^{(i)} = \exp\Big(\tfrac{1}{\tau} f_\psi(\tilde{x}_{t-1}^{(i)}, p)\Big), \tag{15}$$

$$w_{t-1}^{(i)} = \frac{\tilde{w}_{t-1}^{(i)}}{\sum_{j=1}^K \tilde{w}_{t-1}^{(j)}}. \tag{16}$$

The normalized weights $\{w_t^{(i)}\}_{i=1}^K$ concentrate probability mass on high-critic-score particles.

Resampling. Finally, we resample particles according to the normalized weights:

$$a_{t-1}^{(i)} \sim \mathrm{Categorical}\big(w_{t-1}^{(1:K)}\big), \tag{17}$$

$$x_{t-1}^{(i)} = \tilde{x}_{t-1}^{(a_{t-1}^{(i)})}, \tag{18}$$

which duplicates high-weight particles and discards low-weight ones, yielding a new equally weighted set $\{x_t^{(i)}\}_{i=1}^K$ for the next step.

After the final diffusion step, we obtain a set of particles $\{x_0^{(i)}, w_0^{(i)}\}_{i=1}^K$. We use the critic to select the final output as the image with the highest score:

$$i^\star = \arg\max_i f_\psi(x_0^{(i)}), \tag{19}$$

$$\hat{x} = x_0^{(i^\star)}. \tag{20}$$

H Visual Results

Additional Results. We present additional qualitative examples generated by our aligned models in Fig. 7.



Figure 7: Additional visual results generated by our aligned models. All images are produced without classifier-free guidance (CFG).

H.1 Single-Reward and Multi-Reward Qualitative Comparison

Figure 8 compares GenEval-only optimization with the addition of HPS. These examples accompany the discussion in Appendix J. The three-reward quantitative experiment uses CLIP, HPSv2.1, and GenEval and is a separate setting.

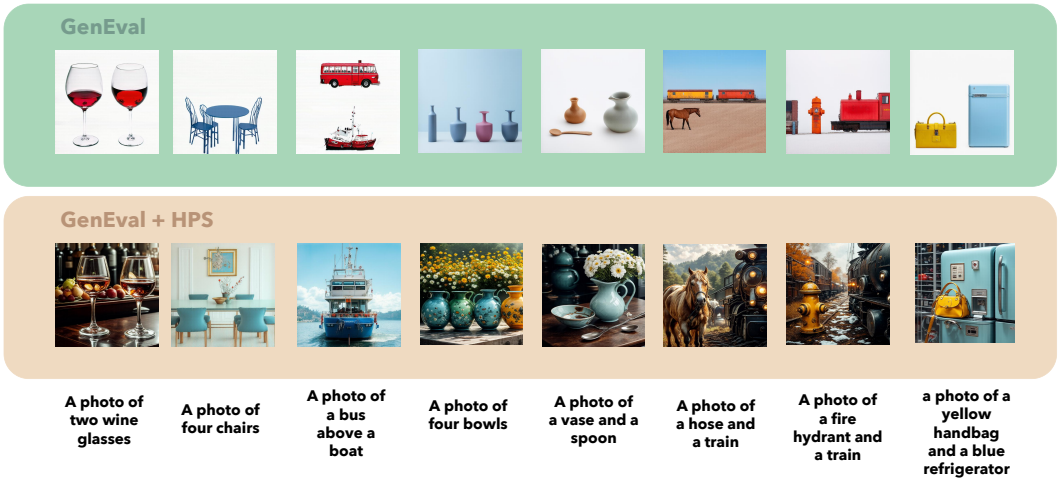

Figure 8: **Qualitative comparison of single-reward and multi-reward optimization.** GenEval-only training (top) and GenEval-plus-HPS training (bottom). The selected examples illustrate differences in composition, background detail, and visual appearance. They do not by themselves establish a general reduction in reward hacking.

Table 7: **Ablation on value pretraining.** Value pretraining improves performance, especially on verifiable rewards.

	vp 15it	vp 50it	vp 100it	w/o vp
GenEval ↑	**0.6392**	0.6138	0.5932	0.5419

Table 8: **Ablation on separate value heads.** Separate heads consistently outperform a shared head across all rewards.

	CLIP ↑	HPSv2.1 ↑	GenEval ↑
Separate head	**0.2896**	**0.3441**	**0.4986**
Shared head	0.2804	0.3138	0.4238

I Additional Ablation Studies

We report supporting ablations on critic warm-up, separate reward heads, timestep conditioning, and classifier-free guidance. Critic-backbone comparisons remain in Sec. 4.2, where they address the choice of value representation.

I.1 Effect of Value Pretraining

Next, we evaluate the effect of the value-pretraining phase. We run this ablation on GenEval, which gives a sparser and less smooth signal than HPSv2.1. We compare two training settings: (i) training PPO directly from the diffusion-initialized critic without value pretraining, and (ii) doing a short value-pretraining stage before enabling actor updates. We keep all other hyperparameters the same. As shown in Table 7, adding value pretraining improves the final performance. This supports our claim that warming up the critic gives the policy better advantage estimates, especially for hard or discrete rewards.

I.2 Separate Heads for Multi-reward Training

We study the critic design for the multi-reward setting, where we share the diffusion backbone but predict each reward with a separate lightweight MLP value head. We compare this design to a shared-head variant that regresses all rewards with a single scalar head. As shown in Table 8, separate heads consistently outperform the shared head across all rewards (CLIP, HPSv2.1, and GenEval), suggesting that decoupling value prediction reduces interference between heterogeneous objectives and better accommodates reward-specific scales and semantics.

I.3 Critic-Head Conditioning

We examine whether the value readout should receive the timestep explicitly, even though the diffusion backbone already uses it. Table 9 isolates time and text conditioning at the critic head.

Without additional text conditioning, adding time conditioning improves the HPSv2.1 score from 0.3496 to 0.3642. With text conditioning, it improves the score from 0.3524 to 0.3630.

Extra text conditioning does not improve the time-conditioned head in this experiment. This supports the timestep-aware readout used in Sec. 3.3.

Table 9: **Ablation on critic-head conditioning.**

Time cond.	Text cond.	HPSv2.1 ↑
✗	✗	0.3496
✗	✓	0.3524
✓	✗	**0.3642**
✓	✓	0.3630

I.4 Effect of CFG on training.

We ablate the effect of classifier-free guidance (CFG) on training and use *CFG-free* training ($\mathrm{cfg} = 1.0$) in all main experiments. Besides reducing memory usage and improving training speed, CFG-free training also yields the best optimization behavior. As shown in Fig. 9, larger CFG scales improve the initial reward, but lead to slower subsequent gains and more unstable reward curves. In particular, higher CFG introduces stronger fluctuations, and $\mathrm{cfg} = 10.0$ even shows reward degradation at later training stages. In contrast, $\mathrm{cfg} = 1.0$ achieves both the most stable trajectory and the highest final reward.

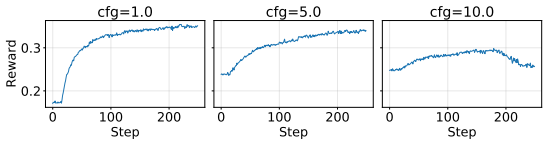

Figure 9: **Effect of CFG on training.** Higher CFG improves the initial reward, but leads to slower and less stable optimization. CFG-free ($\mathrm{cfg} = 1.0$) achieves the best final reward.

J Discussion: Multi-Reward Optimization and Reward Hacking

Accurate value estimation and a well-specified reward address different problems. The critic predicts the expected return under the chosen reward. It does not determine whether that reward captures all aspects of desirable generation. Consequently, improving optimization can also make omissions in a proxy objective more visible. A compositional evaluator, for example, may reward the presence and arrangement of objects while placing little constraint on background detail or visual quality.

Complementary rewards offer a practical way to constrain such omissions. Our shared critic backbone with reward-specific value heads supports this extension without requiring a separate feature extractor for every objective. The GenEval-only and GenEval-plus-HPS examples in Appendix H.1 illustrate changes in background detail and visual appearance when a preference objective is added. They are qualitative evidence consistent with mitigating particular proxy-driven artifacts, rather than a controlled demonstration that reward hacking has been eliminated. The three-reward benchmark in Sec. K.1 separately establishes that the framework can optimize heterogeneous objectives, with measurable trade-offs among them.

A stronger test of mitigation would track the optimized rewards alongside an independent quality or preference assessment over training, using matched prompts and training budgets. Such an evaluation would distinguish broad quality improvements from simply optimizing an additional proxy. Multi-reward training is therefore an extension of the proposed critic framework and a possible mitigation for reward misspecification, while the central contribution remains state-dependent value estimation for optimizing diffusion models.

K Multi-Reward Optimization and Inference-Time Steering

K.1 Multi-Reward Joint Optimization

We next evaluate whether our method remains effective when multiple heterogeneous rewards must be optimized at the same time. From the three categories we pick one representative reward each: CLIP for text-image alignment, HPSv2.1 for human-preference signals, and GenEval for non-differentiable compositional constraints. Training details are provided in Appendix G.

Table 10: Multi-reward optimization.

Model	Method	CLIP ↑	HPSv2.1 ↑	GenEval ↑	Sum ↑
SD1.5	Base	0.2250	0.1727	0.1558	0.5535
	DDPO	0.2590	0.2995	0.3830	0.9415
	DanceGRPO	0.2811	**0.3474**	0.4688	1.0973
	Ours	**0.2896**	0.3441	**0.4986**	**1.1323**
SD3.5-M	Base	0.2275	0.1750	0.2013	0.6038
	DDPO	div	div	div	div
	DanceGRPO	**0.2803**	**0.3392**	0.7705	1.3927
	Ours	0.2772	0.3130	**0.9241**	**1.5143**

Table 10 shows that our method achieves the highest GenEval score and reward sum on both backbones. On SD1.5, it improves CLIP from 0.2811 to 0.2896 and GenEval from 0.4688 to 0.4986 relative to DanceGRPO, with a small HPSv2.1 decrease from 0.3474 to 0.3441.

On SD3.5M, GenEval improves from 0.7705 to 0.9241, while CLIP decreases from 0.2803 to 0.2772 and HPSv2.1 from 0.3392 to 0.3130. The sum increases from 1.3927 to 1.5143. DDPO diverges in this setting.

Thus, joint optimization yields stronger compositional performance, with trade-offs in the other rewards. The sum is scale-dependent and does not imply uniform improvement across objectives. We discuss complementary rewards and proxy overoptimization in Appendix J.

K.2 Inference-Time Steering

We further test whether the critic trained together with PPO can be used purely at inference time to steer sampling and improve rewards, without any extra finetuning. Starting from the post-trained models (SD1.5 and SD3.5-M), we compare four inference-time strategies:

(1) **Best-of-n (BoN):** draw n i.i.d. samples and keep the highest-reward one. (2) **SMC:** following FK-steering (Wu et al., 2023a, Zhang et al., 2025c, Singhal et al., 2025), maintain n particles, evaluate the reward at intermediate timesteps by pushing samples to x_0, and resample according to normalized rewards. (3) **Our Critic Gradient Guidance:** add the critic-derived direction at every denoising step, generating only one sample per prompt. (4) **Critic Gradient Guidance + BoN:** run critic-guided sampling to generate n candidates, then apply BoN selection.

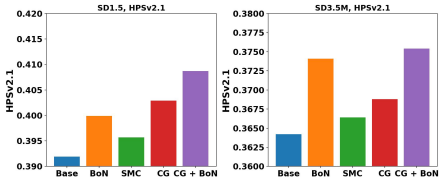

Figure 10: **Inference-time steering** on HPSv2.1 for post-trained SD1.5 / SD3.5-M. We compare BoN, SMC, our critic gradient guidance, and guidance + BoN. Base refers to the RL post-trained model.

We evaluate on HPSv2.1, with additional reward evaluations in Appendix F. Figure 10 shows that critic guidance improves over the post-trained baseline and SMC on both backbones. At the evaluated settings, it also exceeds BoN on SD1.5, while BoN achieves a higher score on SD3.5M.

Combining critic guidance with BoN yields the best performance on both backbones, indicating that trajectory steering and reward-based selection are complementary. Critic guidance uses one trajectory per prompt, whereas BoN and SMC use multiple candidates or particles. This reduces the number of sampled trajectories, although total compute also includes critic evaluation and latent-gradient computation.