RECTIFIED PREFERENCE ALIGNMENT VIA DETERMINISTIC POLICY OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Preference alignment for text-to-image rectified-flow models commonly relies on image-level rewards. However, such supervision is flow-blind: rectified and non-rectified paths to the same preferred image receive identical rewards. Consequently, alignment can improve endpoint reward without preserving the pretrained flow geometry that supports effective finite-step sampling. This may compromise generation quality and weaken alignment gains as the number of sampling steps decreases. We address this gap with Flow-VPA, a Flow-aware Velocity Preference Alignment framework that combines rectified-transport supervision with deterministic policy optimization. To provide this supervision, we introduce BT-Flow, a Bradley–Terry reward model over state–velocity pairs that lifts existing chosen–rejected image comparisons into local velocity preferences. Conditioned on a shared intermediate state, time, and prompt, BT-Flow learns to favor the velocity defining a straight-line completion to the chosen image over the corresponding velocity toward the rejected image. Building on this reward, Flow-VPA treats predicted velocities as deterministic actions and updates the policy using reward gradients at on-policy states. This enables direct alignment of the deterministic flow without introducing an auxiliary stochastic differential equation, evaluating trajectory likelihoods, or differentiating through the sampler. Experiments on SD3.5-M show that Flow-VPA improves preference alignment and generation quality over representative baselines under practical finite-step sampling, with a better trade-off between alignment performance and policy-training compute.

1 INTRODUCTION

Text-to-image rectified-flow models generate images by transporting noise to data along a learned continuous-time velocity field (Lipman et al., 2022; Liu et al., 2022; Esser et al., 2024). Training on straight noise–data interpolations encourages simple transport dynamics, while inference numerically integrates the learned field using a finite number of function evaluations (NFE). Because approximately constant-velocity trajectories can be accurately integrated with few steps, the learned flow geometry is central to the quality–efficiency trade-off in image generation (Liu et al., 2023).

Preference alignment improves generative pre-training by learning from human feedback of generated images (Lee et al., 2023; Liang et al., 2024). A common pipeline first learns an image-level reward from chosen–rejected comparisons using a Bradley–Terry (BT) model (Xu et al., 2023; Kirstain et al., 2023), then fine-tunes the generator against this reward through likelihood-based online reinforcement learning (Black et al., 2024). By optimizing the predicted desirability of output images, this approach can improve aesthetics, prompt adherence, and overall human preference.

However, such image-level supervision is flow-blind: as illustrated in Figure 1(a), a direct rectified path and a non-rectified path to the same preferred image are indistinguishable to the reward. A perfect image evaluator cannot resolve this ambiguity, because the distinction lies in the path rather than the endpoint. The policy can thus pursue higher rewards through routes that depart from pretrained rectification. Losing this structure makes the resulting quality gains harder to retain under coarser sampling. Moving beyond endpoints, state-based rewards (Dai et al., 2025; Liang et al., 2025) provide denser feedback and can distinguish trajectories through the visited intermediate states. Yet individually preferred states need not lie along a common rectified continuation from the current on-policy state, as shown in Figure 1(b). Rewarding these states therefore does not necessarily favor

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.

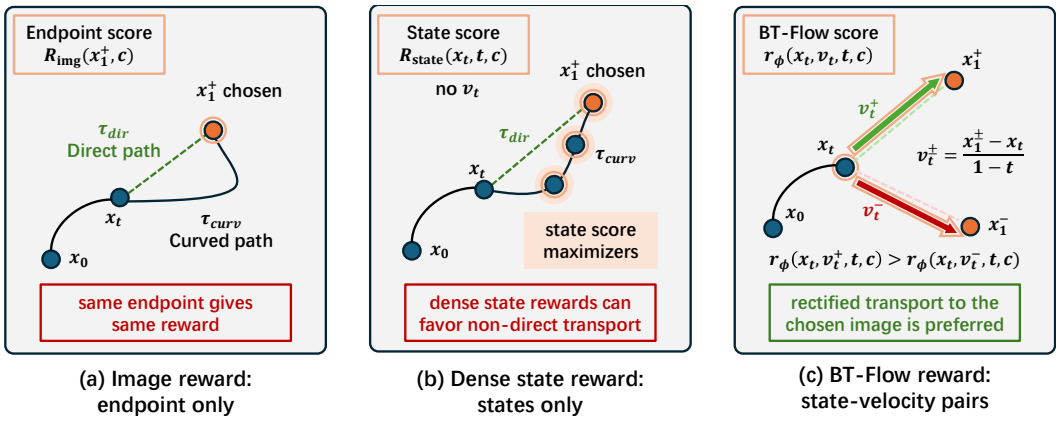


Figure 1: **From flow-blind to flow-aware preference supervision.** (a) Image-level rewards evaluate the preferred endpoint, and therefore cannot distinguish straight and curved transports that reach the same preferred image. (b) State-wise rewards provide denser supervision over intermediate states, but remain flow-blind: the preferred states need not form a direct trajectory. (c) BT-Flow lifts chosen–rejected image preferences to local state–velocity preferences. At a shared state x_t, the velocities $v_t^\pm = (x_1^\pm - x_t)/(1-t)$ define rectified completions to the respective images over the remaining time. The reward $r_\phi(x_t, v_t, t, c)$ is trained to rank the chosen velocity v_t^+ above the rejected velocity v_t^-. Optimizing this reward encourages rectified transport to human-preferred images.

direct transport toward a preferred endpoint. The missing ingredient is a preference signal over local velocity predictions that explicitly favors rectified transport toward preferred images.

We address this gap with Flow-VPA, a Flow-aware Velocity Preference Alignment framework, as shown in Figure 1(c). The core component, BT-Flow, is a Bradley–Terry Flow reward model over state–velocity pairs that lifts existing image comparisons into local velocity preferences. Given a chosen–rejected image pair and a shared noise sample, we construct intermediate states along the two noise–image interpolations. At each state and time, the chosen and rejected endpoints define two candidate velocities, each of which would reach its corresponding image through constant-velocity transport over the remaining time. Conditioned on the same state, time, and prompt, BT-Flow learns to rank the chosen-directed velocity above its rejected-directed counterpart. This construction couples endpoint preferences with rectified local transport without additional human preference data. Our theoretical analysis formalizes this connection: in an idealized alignment setting, the canonical flow objective selects exactly those image-alignment optima that realize direct rectified transport.

Building on BT-Flow, Flow-VPA treats predicted velocities as deterministic actions and optimizes them using the learned Flow-RM. At a fixed state collected by an on-policy ODE rollout, the reward expectation over actions reduces to a single evaluation at the predicted velocity. Differentiating this reward through the velocity field yields a direct policy gradient, enabling alignment of the deterministic flow without introducing an auxiliary SDE. The resulting framework is likelihood-free, with policy updates that are agnostic to the choice of ODE solver. Experiments on SD3.5-M demonstrate improved preference alignment and generation quality under practical finite-step sampling over representative methods spanning online alignment with image-level or state-level rewards. Beyond these quality gains, Flow-VPA offers a more favorable trade-off between alignment performance and policy-training compute than the evaluated online baselines.

To summarize, our contributions are as follows:

- We identify and characterize flow-blindness, showing why image-level and state-level preference supervision does not by itself ensure rectified transport.
- We propose BT-Flow, which lifts existing image comparisons into preference-preserving velocity supervision without additional human annotations. Under stated assumptions, we prove that ideal flow alignment strictly refines the image-alignment solution set by retaining exactly the optima inducing direct rectified transport.

- We introduce Flow-VPA, a likelihood-free algorithm that enables direct ODE alignment through deterministic velocity optimization, without an auxiliary SDE.
- We demonstrate improved preference alignment and generation quality, better retention of alignment gains at lower NFE, and a more favorable trade-off between alignment performance and policy-training compute.

2 Related Work

Preference Modelling for Image Generation has expanded from evaluating completed images to assessing intermediate generation states. ImageReward (Xu et al., 2023), PickScore (Kirstain et al., 2023), and HPSv2 (Wu et al., 2023) learn prompt-conditioned image scores from human comparisons, supporting rejection sampling, model alignment and evaluation. This image-level formulation accommodates different backbones: HPSv3 (Ma et al., 2025) builds on a pretrained vision–language model, whereas DiT-Reward (Yang et al., 2026) extracts preference signals from pretrained diffusion transformers. Beyond endpoint evaluation, SPO (Liang et al., 2025) introduces a step-aware preference model, while LRM (Zhang et al., 2026) repurposes diffusion-model features to score noisy latents directly. Unlike these models scoring images or latent states, BT-Flow lifts endpoint state comparisons to learn a Bradley–Terry reward over state–velocity pairs.

Preference Alignment for Diffusion and Flow Models translates human preferences into supervision for a multi-step generative process. Direct preference objectives learn from image comparisons through denoising losses, with step-wise variants localizing supervision to individual transitions (Wallace et al., 2024; Liang et al., 2025; Zhang et al., 2026). With explicit rewards, methods either backpropagate image-level feedback through sampling chains (Clark et al., 2024) or treat denoising as sequential decision-making and apply policy-gradient updates (Fan et al., 2023; Black et al., 2024; Xue et al., 2025b; He et al., 2026a; Liu et al., 2026a). Within the latter framework, denser temporal rewards further improve credit assignment (Deng et al., 2026; He et al., 2026b). Whereas these policy-gradient methods update stochastic or surrogate policies, Flow-VPA directly aligns local velocity actions through deterministic policy optimization under BT-Flow.

Forward-Process Alignment offers a complementary route by incorporating preference feedback into flow-matching objectives on re-noised samples. Pairwise formulations such as Flow-DPO encode endpoint comparisons through reference-relative velocity-prediction errors (Liu et al., 2026b). Reward-based formulations instead emphasize high-scoring samples through reward-weighted flow-matching regression (Fan et al., 2025b; Liu et al., 2026b). The weighting signal can be refined using group-relative advantages or intermediate-state critics (Xue et al., 2025a; Fan et al., 2025a). Beyond sample reweighting, negative-aware formulations incorporate favourable and unfavourable outcomes through implicit positive and negative velocity fields (Zheng et al., 2026). Further refinements stabilize signed-advantage regression and construct advantage-dependent velocity targets (Kveton et al., 2026; Fu et al., 2026). Even though their velocity supervision is applied to re-noised states, such a distribution need not match that encountered during sampling. To fill this gap, Flow-VPA instead aligns transport decisions at intermediate states visited by the current policy.

3 Preliminaries

We introduce the notation for pairwise preference modeling, conditional rectified flow, finite-step sampling, and likelihood-based online alignment. Throughout, $t = 0$ denotes the Gaussian source and $t = 1$ denotes the data endpoint.

Bradley–Terry preference modeling. Under the Bradley–Terry model (Bradley & Terry, 1952), pairwise human preferences are parameterized by a latent scalar reward function $r^\star$. Given a prompt $\boldsymbol{c}$ and two endpoint images $\boldsymbol{x}_1^+$ and $\boldsymbol{x}_1^-$, the human preference probability is modeled as

$$p^\star\left(\boldsymbol{x}_1^+ \succ \boldsymbol{x}_1^- \mid \boldsymbol{c}\right) = \sigma\left(r^\star\left(\boldsymbol{x}_1^+, \boldsymbol{c}\right) - r^\star\left(\boldsymbol{x}_1^-, \boldsymbol{c}\right)\right), \tag{1}$$

where $\sigma(\cdot)$ denotes the logistic function. Let $\mathcal{D}$ be a human-labeled preference dataset whose tuples $(\boldsymbol{c}, \boldsymbol{x}_1^+, \boldsymbol{x}_1^-)$ record that $\boldsymbol{x}_1^+$ was preferred to $\boldsymbol{x}_1^-$. An image-level reward model $r_\phi^{\text{img}}(\boldsymbol{x}_1, \boldsymbol{c})$ is trained as a proxy for the latent reward $r^\star$ by minimizing the Bradley–Terry loss (Xu et al., 2023)

$$\mathcal{L}_{\text{BT}}^{\text{img}}(\phi) = -\mathbb{E}_{(\boldsymbol{c}, \boldsymbol{x}_1^+, \boldsymbol{x}_1^-) \sim \mathcal{D}}\left[\log \sigma\left(r_\phi^{\text{img}}\left(\boldsymbol{x}_1^+, \boldsymbol{c}\right) - r_\phi^{\text{img}}\left(\boldsymbol{x}_1^-, \boldsymbol{c}\right)\right)\right]. \tag{2}$$

Thus, r_ϕ^{img} is trained to rank clean image endpoints at $t = 1$.

Conditional rectified flow. We consider a conditional rectified flow that transports a Gaussian source distribution $p_0 = \mathcal{N}(\mathbf{0}, \mathbf{I})$ to the conditional data distribution $p_{\text{data}}(\cdot \mid \boldsymbol{c})$. Given $\boldsymbol{x}_0 \sim p_0$, $\boldsymbol{x}_1 \sim p_{\text{data}}(\cdot \mid \boldsymbol{c})$, and $t \in [0, 1]$, rectified flow constructs the linear interpolation (Liu et al., 2022)

$$\boldsymbol{x}_t = (1 - t)\boldsymbol{x}_0 + t\boldsymbol{x}_1, \qquad \boldsymbol{v}_t = \boldsymbol{x}_1 - \boldsymbol{x}_0. \tag{3}$$

For each sampled source–data pair, Eq. 3 specifies a straight path with a constant target velocity. A conditional velocity field $\boldsymbol{v}_\theta(\boldsymbol{x}_t, t, \boldsymbol{c})$ is trained to predict this target by minimizing the flow-matching objective (Lipman et al., 2022)

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{\boldsymbol{c}, \boldsymbol{x}_0 \sim p_0, \boldsymbol{x}_1 \sim p_{\text{data}}(\cdot|\boldsymbol{c}), t \sim \mathcal{U}[0,1]} \left[\left\| \boldsymbol{v}_\theta\left(\boldsymbol{x}_t, t, \boldsymbol{c}\right) - \boldsymbol{v}_t \right\|_2^2 \right]. \tag{4}$$

Finite-step sampling. At inference, the learned velocity field induces a continuous model trajectory

$$\frac{\mathrm{d}\boldsymbol{x}_t^\theta}{\mathrm{d}t} = \boldsymbol{v}_\theta\left(\boldsymbol{x}_t^\theta, t, \boldsymbol{c}\right), \qquad \boldsymbol{x}_0^\theta = \boldsymbol{x}_0. \tag{5}$$

Since this trajectory is generally unavailable in closed form, samples are generated by numerical integration. On a uniform time grid $k = 0, \Delta t, \ldots, 1 - \Delta t$, with $\Delta t = 1/N$, the N-step Euler sampler performs

$$\boldsymbol{x}_N^{(k+\Delta t)} = \boldsymbol{x}_N^{(k)} + \Delta t\, \boldsymbol{v}_N^{(k)}, \qquad \boldsymbol{v}_N^{(k)} = \boldsymbol{v}_\theta\left(\boldsymbol{x}_N^{(k)}, k, \boldsymbol{c}\right). \tag{6}$$

The resulting finite-step trajectory is

$$\tau_N = \left(\boldsymbol{x}_N^{(0)}, \boldsymbol{v}_N^{(0)}, \boldsymbol{x}_N^{(\Delta t)}, \boldsymbol{v}_N^{(\Delta t)}, \ldots, \boldsymbol{x}_N^{(1)}\right), \qquad \boldsymbol{x}_N^{(0)} = \boldsymbol{x}_0. \tag{7}$$

For Euler sampling, N is also the number of function evaluations (NFE). The numerical endpoint $\boldsymbol{x}_N^{(1)}$ approximates $\boldsymbol{x}_1^\theta$ and is returned as the generated sample. Under standard regularity conditions, the numerical trajectory converges to the continuous trajectory as $N \to \infty$ (Driscoll & Braun, 2017).

At finite N, discretization error depends on both the step size and the variation of the learned velocity field along the model trajectory. When the velocity varies slowly along the trajectory, coarse time discretization can still yield accurate samples, supporting low-NFE generation (Liu et al., 2022).

Likelihood-based alignment. Transition-level likelihood-based optimization requires tractable transition densities (Black et al., 2024), whereas the Euler update in Eq. 6 is deterministic. Flow-GRPO (Liu et al., 2026a) introduces stochastic transitions through an ODE-to-SDE construction designed to preserve continuous-time marginals, followed by Euler–Maruyama discretization. Under its isotropic Gaussian construction, each one-step transition admits a tractable density. For these stochastic rollouts, the likelihood of the state sequence in τ_N factorizes as

$$P_\theta\left(\tau_N \mid \boldsymbol{x}_N^{(0)}, \boldsymbol{c}\right) = \prod_{k \in \{0, \Delta t, \ldots, 1-\Delta t\}} p_\theta\left(\boldsymbol{x}_N^{(k+\Delta t)} \mid \boldsymbol{x}_N^{(k)}, k, \boldsymbol{c}\right). \tag{8}$$

Here, p_θ denotes the one-step transition density, and P_θ denotes the induced trajectory distribution; both are parameterized through the velocity field $\boldsymbol{v}_\theta$. Using the learned image-level reward r_ϕ^{img}, we write the reference-regularized image-alignment objective as

$$\mathcal{J}_{\text{ImageAlign}}(\theta; r_\phi^{\text{img}}) = \mathbb{E}_{\boldsymbol{c}, \boldsymbol{x}_0 \sim p_0, \tau_N \sim P_\theta(\cdot|\boldsymbol{x}_0, \boldsymbol{c})} \left[r_\phi^{\text{img}}\left(\boldsymbol{x}_N^{(1)}, \boldsymbol{c}\right) \right] - \beta \mathbb{D}_{\text{KL}}\left[p_\theta \| p_{\text{ref}}\right], \tag{9}$$

where p_{ref} is induced by a fixed reference model, $\mathbb{D}_{\text{KL}}[p_\theta \| p_{\text{ref}}]$ denotes the expected transition-averaged KL divergence, and $\beta \geq 0$ controls the regularization strength. The reference penalty discourages deviations from the reference transitions, mitigating reward over-optimization that exploits errors in the learned approximation to the latent reward (Gao et al., 2023; Liu et al., 2026a).

4 Why Endpoint Preference Alignment Is Flow-Blind

Existing image-level alignment methods for flow models evaluate human preference at $\boldsymbol{x}_N^{(1)}$, the numerical endpoint of a sampled trajectory. However, endpoint supervision does not directly evaluate the transport dynamics that produce the final sample, a limitation we refer to as flow-blindness. In this section, we first formalize the resulting ambiguity in local velocity predictions and then examine how it affects finite-step sampling as NFE is reduced. Appendix C further extends the theoretical analysis to state-based preference alignment, which can constrain state visitation but does not directly evaluate candidate velocities at a given state and time.

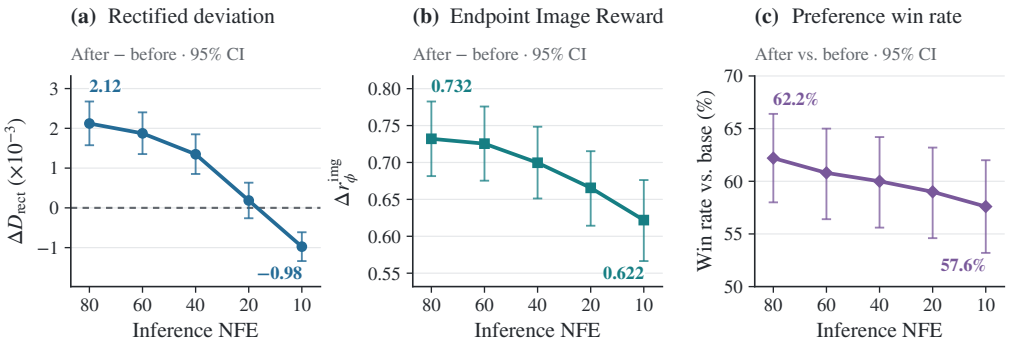


Figure 2: **Effects of endpoint-reward alignment across inference budgets.** (a) Change in rectified path deviation, ΔD_{rect}. This deviation measures departure from constant-velocity interpolation between each trajectory's endpoints. (b) Change in endpoint image reward, $\Delta r_\phi^{\text{img}}$. Both changes are computed as after minus before alignment. (c) Preference win rate of the aligned model over the base model, judged by GPT-5.6 Luna. Error bars indicate 95% confidence intervals.

4.1 Endpoint Rewards Underdetermine Transport Dynamics

Although a terminal reward can be used to update every transition, it does not directly specify which local velocity should be preferred at each step. Under Euler integration, the numerical endpoint depends on the accumulated velocity predictions along the trajectory. Different local velocity sequences can therefore produce the same endpoint and receive the same image-level reward. The following proposition formalizes this velocity non-identifiability, and the proof is in Appendix B.1.

Proposition 4.1 *For a fixed prompt $\mathbf{c}$, initial state $\boldsymbol{x}_0$, and sampling budget $N \geq 2$, let τ_N be an Euler trajectory with uniform step size $\Delta t = 1/N$. Construct a second Euler trajectory from the same initial state by applying $\{\boldsymbol{\delta}_N^{(0)}, \ldots, \boldsymbol{\delta}_N^{(1-\Delta t)}\}$ to the realized velocity of τ_N at each step. The two trajectories have the same image-level reward, whenever $\sum_{k\in\{0,\Delta t,\ldots,1-\Delta t\}} \boldsymbol{\delta}_N^{(k)} = \mathbf{0}$.*

Proposition 4.1 shows that endpoint preference supervision cannot distinguish local velocity sequences that produce the same endpoint. It evaluates these sequences only through the aggregate displacement that determines the final sample. Thus, a direct trajectory and one containing substantial intermediate detours can be indistinguishable under the image-level preference term. Endpoint rewards therefore supervise where the trajectory ends, but underdetermine how it arrives there.

This ambiguity is particularly relevant to rectified-flow models. The flow-matching objective in Eq. 4 is constructed from straight conditional paths with constant target velocities, reflecting the goal of efficient few-step sampling through approximately rectified transport. The image-level preference term, however, does not explicitly enforce preservation of this local transport structure during fine-tuning. Consequently, trajectories that are equivalent under the endpoint reward at a fixed NFE budget may differ in their robustness to coarser numerical integration.

4.2 Empirical Analysis of Alignment and NFE Sensitivity

To examine the practical implications of flow-blind supervision, we align SD3.5-M using Flow-GRPO with an image reward model r_ϕ^{img} trained on Pick-a-Pic v2. We evaluate the pretrained and aligned models at matched NFE budgets using the same held-out prompts, initial noises, deterministic sampler, and guidance scale. Metric definitions and experimental details are in Appendix E.5.

Endpoint alignment changes sampled flow geometry. Figure 2(a) reports the change in rectified path deviation, which measures departures from constant-velocity interpolation between each numerical trajectory's own endpoints on its native sampling grid. As NFE decreases from 80 to 10, the aligned-minus-base difference decreases overall from 2.12×10^{-3} to -0.98×10^{-3}, becoming slightly negative at 10 NFE. Alignment thus alters the sampled transport geometry, with greater excess rectified deviation relative to the pretrained model at higher NFE. These changes are consistent

with Proposition 4.1: endpoint rewards leave local velocities underdetermined, so flow-blind optimization can reshape the pretrained model's transport geometry without explicitly supervising how preferred outcomes are reached.

Alignment gains diminish as NFE decreases. Alongside these transport changes, the aligned model's advantage over the matched-NFE pretrained model becomes smaller at lower inference budgets, as shown in Figure 2(b,c). From 80 to 10 NFE, the held-out image-reward gain $\Delta r_\phi^{\text{img}}$ falls from 0.732 to 0.622, while the GPT-5.6 Luna preference win rate declines overall from 62.2% to 57.6%. Stronger alignment gains thus coexist with greater excess rectified deviation at higher NFE. Conversely, the slightly lower deviation relative to the pretrained model at 10 NFE is accompanied by smaller gains in both the image reward and external preference judgments. Although the aligned model remains more preferable, its stronger observed gains come at the cost of additional NFE. This exposes a practical limitation of endpoint-only supervision: improving final preferences does not ensure that the learned transport can retain those improvements under finite-step sampling.

5 Flow-Aware Velocity Preference Alignment

As terminal rewards solely supervise the final sample but not the local transport that produces it, endpoint-based alignment can fail to preserve its preference gains under empirical inference NFE. We address this limitation with a two-stage framework. First, BT-Flow lifts endpoint preference pairs into state-conditioned velocity comparisons. Second, Flow-VPA optimizes the velocity field at on-policy states through likelihood-free deterministic policy optimization. Practical implementation details and pseudocode are provided in Appendix D.

5.1 BT-Flow Preference Modeling

Ideally, we aim to lift an endpoint preference pair $(\boldsymbol{c}, \boldsymbol{x}_1^+, \boldsymbol{x}_1^-)$ at a shared on-policy state $\boldsymbol{x}_t^\theta$ visited by the current flow model. More generally, given any shared intermediate state $\boldsymbol{x}_t$ and time $t < 1$, we construct two direct completion velocities:

$$\boldsymbol{v}_t^+ = \frac{\boldsymbol{x}_1^+ - \boldsymbol{x}_t}{1-t}, \qquad \boldsymbol{v}_t^- = \frac{\boldsymbol{x}_1^- - \boldsymbol{x}_t}{1-t}. \tag{10}$$

Each velocity completes the remaining transport from the current state to its corresponding endpoint: $\boldsymbol{x}_t + (1-t)\boldsymbol{v}_t^\pm = \boldsymbol{x}_1^\pm$. Because the state, time, and prompt are shared, the comparison isolates the preferred velocity rather than confounding velocity preference with differences between intermediate states. We lift the endpoint preference to the state-conditioned velocity comparison $\boldsymbol{v}_t^+ \succ \boldsymbol{v}_t^-$ and model its probability using the Bradley–Terry likelihood:

$$p^\star_{\text{flow}}\left(\boldsymbol{v}_t^+ \succ \boldsymbol{v}_t^- \mid \boldsymbol{x}_t, t, \boldsymbol{c}\right) = \sigma\left(r^\star_{\text{flow}}\left(\boldsymbol{x}_t, \boldsymbol{v}_t^+, t, \boldsymbol{c}\right) - r^\star_{\text{flow}}\left(\boldsymbol{x}_t, \boldsymbol{v}_t^-, t, \boldsymbol{c}\right)\right). \tag{11}$$

This construction provides a transition-level preference: from the same intermediate state, direct completion toward the chosen endpoint is preferred to direct completion toward the rejected endpoint. Constructing these comparisons from on-policy states would require repeated model rollouts throughout reward-model training. We therefore sample uniformly from synthetic states:

$$\boldsymbol{x}_t \sim \text{Unif}(X_t), \qquad X_t = \left\{(1-t)\boldsymbol{x}_0 + t\boldsymbol{x}_1^+,\ (1-t)\boldsymbol{x}_0 + t\boldsymbol{x}_1^-\right\}.$$

We then train the flow-aware reward model $r_\phi^{\text{flow}}(\boldsymbol{x}_t, \boldsymbol{v}_t, t, \boldsymbol{c})$ using the BT-Flow objective

$$\mathcal{L}_{\text{BT}}^{\text{flow}}(\phi) = -\mathbb{E}_{(\boldsymbol{c}, \boldsymbol{x}_1^+, \boldsymbol{x}_1^-)\sim\mathcal{D}, \boldsymbol{x}_0, t, \boldsymbol{x}_t}\left[\log\sigma\left(r_\phi^{\text{flow}}\left(\boldsymbol{x}_t, \boldsymbol{v}_t^+, t, \boldsymbol{c}\right) - r_\phi^{\text{flow}}\left(\boldsymbol{x}_t, \boldsymbol{v}_t^-, t, \boldsymbol{c}\right)\right)\right]. \tag{12}$$

5.2 Flow-VPA via Deterministic Policy Optimization

After training, we freeze the BT-Flow reward model and use it to align the velocity predictor. To extend Eq. 9 from endpoint supervision to transition-level flow supervision, we replace the terminal image reward with BT-Flow reward evaluated on an on-policy state–velocity pair:

$$\mathcal{J}_{\text{FlowAlign}}(\theta; r_\phi^{\text{flow}}) = \mathbb{E}_{\boldsymbol{c}, \boldsymbol{x}_0, \boldsymbol{x}_N^{(k)}\sim d_N^\theta, \boldsymbol{v}_t\sim p_\theta(\cdot\mid\boldsymbol{x}_N^{(k)}, k, \boldsymbol{c})}\left[r_\phi^{\text{flow}}\left(\boldsymbol{x}_N^{(k)}, \boldsymbol{v}_t, k, \boldsymbol{c}\right)\right] - \beta\mathbb{D}_{\text{KL}}[p_\theta \| p_{\text{ref}}], \tag{13}$$

where d_N^θ denotes the state-visitation distribution induced by the current velocity field and NFE budget N. Because both the velocity predictor and the pre-trained reward model are differentiable, Flow-VPA instantiates this objective in Eq. 13 using the deterministic model prediction as the action. Thus, we replace the sampled velocity with $\boldsymbol{v}_\theta(\boldsymbol{x}_N^{(k)}, k, \boldsymbol{c})$ and optimize

$$\mathcal{J}_{\text{FlowVPA}}(\theta) = \mathbb{E}_{\boldsymbol{c},\boldsymbol{x}_0,\boldsymbol{x}_N^{(k)}\sim d_N^\theta}\left[r_\phi^{\text{flow}}\left(\boldsymbol{x}_N^{(k)}, \boldsymbol{v}_\theta\left(\boldsymbol{x}_N^{(k)}, k, \boldsymbol{c}\right), k, \boldsymbol{c}\right)\right] - \beta \left\|\boldsymbol{v}_\theta, \boldsymbol{v}_{\text{ref}}\right\|_2^2, \tag{14}$$

where $\|\boldsymbol{v}_\theta, \boldsymbol{v}_{\text{ref}}\|_2^2$ denotes the squared ℓ_2 distance between the current and reference velocity predictions at the same visited state. When moving from stochastic to deterministic policy optimization, this reference velocity penalty replaces the distributional KL penalty and constrains the aligned velocity field toward the reference predictor. During optimization, gradients through d_N^θ are stopped, and the expectation is estimated using states sampled from the current on-policy state dataset $\mathcal{D}_\tau$.

Flow-VPA treats numerical integration as the environment transition and does not model its transition probabilities. Rollouts can therefore be collected using any numerical solver. Moreover, it requires no transition log probabilities and avoids the auxiliary stochastic transition process, which makes Flow-VPA an effectively likelihood-free alignment method.

5.3 Theoretical Analysis

We establish two theoretical properties that characterize the advantages of Flow-VPA over image-based alignment. The proofs are provided in Appendices B.2 and B.3.

Preference-preserving endpoint-to-flow lifting. BT-Flow preference modeling admits a canonical population-level reward that preserves the original Bradley–Terry endpoint preferences.

Proposition 5.1 *Suppose endpoint preferences follow the Bradley–Terry model in Eq. 1. Define the canonical latent flow reward $r_{\text{flow}}^\star(\boldsymbol{x}, \boldsymbol{v}, t, \boldsymbol{c}) = r^\star(\boldsymbol{x} + (1-t)\boldsymbol{v}, \boldsymbol{c})$. Then, for every shared state $\boldsymbol{x}_t$, time $t < 1$, and endpoint pair $(\boldsymbol{x}_1^+, \boldsymbol{x}_1^-)$, the velocities constructed in Eq. 10 satisfy*

$$p_{\text{flow}}^\star\left(\boldsymbol{v}_t^+ \succ \boldsymbol{v}_t^- \mid \boldsymbol{x}_t, t, \boldsymbol{c}\right) = p^\star\left(\boldsymbol{x}_1^+ \succ \boldsymbol{x}_1^- \mid \boldsymbol{c}\right).$$

Proposition 5.1 shows that BT-Flow admits a population-level solution that is exactly preference-equivalent to the endpoint reward. The lifting therefore preserves endpoint preference probabilities while expressing them as comparisons between direct-completion velocities at a shared state.

Strict refinement by direct rectified transport. Even with exact preference rewards, maximizing endpoint preference does not determine how the preferred endpoint is reached. We characterize this structural ambiguity through the unregularized objectives under $r^\star$ and its induced flow reward $r_{\text{flow}}^\star$.

Theorem 5.1 *Fix a source noise $\boldsymbol{x}_0$, a prompt $\boldsymbol{c}$, and $N \geq 2$ Euler steps. Suppose the latent reward $r^\star(\cdot, \boldsymbol{c})$ has a unique maximizer $\boldsymbol{x}_1^\star$. Consider the ideal image and flow alignment objectives in Eqs. 9 and 13, using $r^\star$ and its canonical lift $r_{\text{flow}}^\star$, without reference regularization. Denote their respective maximizer sets by $\Theta_{\text{Image}}^\star$ and $\Theta_{\text{Flow}}^\star$. Whenever $\Theta_{\text{Image}}^\star$ admits both direct and non-direct trajectories from $\boldsymbol{x}_0$ to $\boldsymbol{x}_1^\star$, flow alignment strictly refines image alignment:*

$$\boxed{\Theta_{\text{Flow}}^\star = \Theta_{\text{Image}}^\star \cap \{\theta : \theta \textit{ induces a direct rectified rollout}\} \subsetneq \Theta_{\text{Image}}^\star.}$$

Theorem 5.1 gives an exact characterization of this parameter-set refinement. Among trajectories terminating at the same preferred endpoint, the image-level reward is identical and therefore provides no human-preference signal for distinguishing their transport geometry. Flow alignment retains the preferred endpoint while imposing direct rectified transport as an additional optimality condition.

Taken together, Proposition 5.1 and Theorem 5.1 show that BT-Flow preserves *what* humans prefer, while flow-aware alignment strictly refines the corresponding optimal parameter set by constraining *how* the preferred endpoint is reached.

6 Experiments

We evaluate Flow-VPA on SD3.5-M (Esser et al., 2024) to assess human-preference alignment under practical sampling budgets. We first train our flow-aware reward model, Flow-RM, on 851.3K

Table 1: **Overall preference-alignment results on SD3.5-M.** All models are evaluated at 40 NFE on 500 held-out Pick-a-Pic v2 prompts. Eval. Reward is scored by Image-RM; GPT-5.6 Luna and Terra judge pairwise win rates against SD3.5-M (Base) under matched sampling settings. For Qwen-Image-Bench, QAA-F averages over quality, aesthetics, alignment, and fidelity, while All5 additionally includes creative generation. Higher is better for all metrics. The best is highlighted.

RM Type	Algorithm	Eval.	Ref. Win rate (%)		Qwen-Image-Bench	
		Reward	GPT-5.6 Luna	GPT-5.6 Terra	QAA-F	All5
	Base	0.407	—	—	46.93	44.41
Image-RM	Flow-DPO	0.620	53.40	56.00	47.21	44.72
	Flow-GRPO	1.106	60.00	57.60	48.26	45.51
State-RM	Flow-GRPO	1.768	69.40	60.20	50.16	48.07
	Flow-LPO	1.923	65.20	57.00	49.10	47.16
Flow-RM	**Flow-VPA**	2.017	72.80	65.40	51.39	49.63

pairwise comparisons from Pick-a-Pic v2 (Kirstain et al., 2023) using our BT-Flow formulation. We then align SD3.5-M with Flow-VPA on deduplicated prompts from the same dataset, using Flow-RM for reward supervision and LoRA (Hu et al., 2021) ($r = 32$, $\alpha = 64$) for policy adaptation. The pretrained SD3.5-M serves as the reference model. During online alignment, we collect on-policy trajectories using a first-order ODE solver with 10 NFE and a classifier-free guidance scale of 4.5.

We evaluate at 40 NFE on 500 validation prompts held out from both reward-model training and policy alignment. For each prompt, we generate aligned and reference images from the same initial noise, using the same ODE solver and classifier-free guidance scale. We report three complementary metrics: (i) in-domain reward scores from an image-level reward model (Image-RM) trained on the same 851.3K comparisons; (ii) pairwise preference win rates against the reference model, judged by GPT-5.6 (OpenAI, 2026); and (iii) scores from Q-Judger (Li et al., 2026), the evaluator from Qwen-Image-Bench, covering quality, aesthetics, alignment, real-world fidelity, and creative generation. Full training and evaluation details are provided in Appendix E.

6.1 Results

Flow-VPA outperforms the evaluated baselines across metrics. Table 1 shows that Flow-VPA outperforms the evaluated image-level and state-level online alignment baselines. It achieves pairwise win rates of 72.8% and 65.4% against pretrained SD3.5-M under GPT-5.6 Luna and Terra, respectively, exceeding the strongest baseline under each judge by 3.4 and 5.2 percentage points. Flow-VPA also attains the highest Image-RM reward and Q-Judger scores, with improvements spanning in-domain reward evaluation, pairwise preference judgments, and dimension-wise assessment.

Flow-VPA retains its preference advantage at low NFE. We next revisit the diminishing low-NFE alignment gains identified in our problem analysis. As shown in Figure 3(a), the two state-reward baselines generally achieve higher Luna win rates than image-reward Flow-GRPO, but their preference advantages still diminish as NFE decreases. Thus, denser state-level feedback improves alignment without fully resolving low-NFE attenuation in these experiments. In contrast, Flow-VPA maintains a win rate of at least 70% against the matched-NFE pretrained reference from 80 to 10 NFE, outperforming all evaluated baselines at every budget. Its preference advantage therefore persists under coarser sampling. This contrast is consistent with our analysis: preferred intermediate states need not form a rectified path from the sampled source noise, whereas BT-Flow explicitly favors rectified local transport toward preferred endpoints.

Flow-VPA combines stronger alignment with lower online training cost. Figure 3(b) compares Luna win rates with the computational cost of one epoch of online alignment. Using six rollouts per prompt, Flow-VPA achieves the highest win rate at approximately 170 H100 GPU-hours per epoch, the lowest measured cost among the compared methods. Its cost is comparable to Flow-LPO and substantially lower than that of either Flow-GRPO variant. These results support direct ODE policy

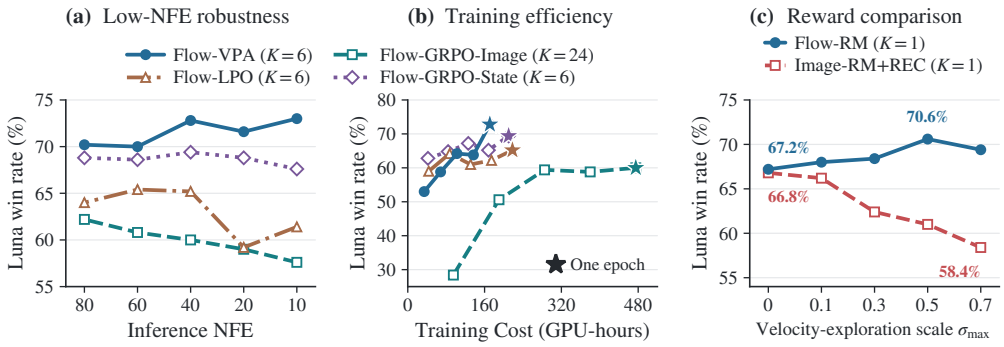


Figure 3: **Low-NFE robustness, online training efficiency, and reward-model ablation.** **(a)** Preference win rates across inference budgets of 10–80 NFE. **(b)** Win rates versus cumulative online training cost. Points mark checkpoints evaluated every 0.2 epochs. Cost is reported in total GPU-hours on four H100 GPUs, excluding reward-model training and held-out evaluation. **(c)** Flow-VPA using Flow-RM versus Image-RM+REC across training-time velocity-exploration scales $\sigma_{\max}$. Rectified endpoint construction (REC) evaluates Image-RM at $\boldsymbol{x}_t + (1-t)\boldsymbol{v}_t$. In the legends, K denotes the number of on-policy rollouts per prompt during online alignment. All win rates are measured against SD3.5-M on 500 held-out prompts using GPT-5.6 Luna as the judge. Paired images share the same initial noise, solver, guidance scale, and NFE. Panels **(b,c)** use 40 NFE.

optimization as an efficient alternative to the evaluated SDE-based alignment methods, achieving stronger preference alignment at lower online training cost.

Explicit state–velocity reward modeling enables more effective exploration. We compare Flow-RM with Image-RM+REC, which evaluates state–velocity pairs using the reconstructed-endpoint reward $r_\phi^{\text{img}}(\boldsymbol{x}_t + (1-t)\boldsymbol{v}_t, \boldsymbol{c})$. Both formulations provide velocity-dependent rewards, so this ablation tests the benefit of explicitly learning state–velocity rewards rather than applying the endpoint-to-flow lift to an image reward. We run Flow-VPA with each reward under otherwise matched settings and vary the velocity-exploration scale $\sigma_{\max}$. Figure 3(c) shows similar performance without exploration: 67.2% Luna win rate for Flow-RM and 66.8% for Image-RM+REC. However, their responses to broader exploration diverge. Flow-RM improves to a peak of 70.6% at $\sigma_{\max} = 0.5$, whereas Image-RM+REC steadily deteriorates, reaching 58.4% at $\sigma_{\max} = 0.7$. These results indicate that the practical advantage of Flow-RM extends beyond velocity dependence alone: it allows Flow-VPA to benefit from broader velocity exploration, while Image-RM+REC loses alignment quality under the same exploration sweep.

7 Conclusion

We identified flow-blindness in preference alignment for rectified-flow models: endpoint rewards cannot distinguish rectified from non-rectified trajectories reaching the same preferred image. To address this ambiguity, we introduced Flow-VPA, which combines BT-Flow reward modeling with deterministic velocity optimization. BT-Flow lifts existing human image comparisons into shared-state velocity preferences, linking preferred endpoints to rectified local transport without additional human annotations. Our theoretical analysis shows that, under idealized assumptions, the flow-aware objective strictly refines the endpoint-alignment solution set, retaining exactly those optima that induce direct rectified transport. Using the learned state–velocity reward, Flow-VPA directly optimizes predicted velocities at on-policy states, enabling ODE-based alignment without an auxiliary SDE, trajectory likelihoods, or differentiation through the sampler. Experiments on SD3.5-M demonstrate improved preference alignment and generation quality over the evaluated baselines. Flow-VPA maintains a strong preference advantage over the pretrained model at matched sampling budgets down to 10 NFE and offers a better trade-off between alignment performance and policy-training compute than the evaluated online baselines. Together, these findings highlight the value of supervising not only which images are preferred, but also how they are reached, supporting effective and efficient alignment under practical finite-step sampling.

AI USE STATEMENT

In this work, we used generative AI tools to assist in the polishing of proofs and evaluate the model alignment results. We have not used generative AI tools for the rest of the required disclosure tasks. Additionally, we used generative AI tools for summarizing existing literature, improving manuscript readability, drafting parts of the appendix, and adapting code for baseline methods. We reviewed all AI-assisted content, checked the revised proofs, assessed the reliability of the AI-based evaluations, and verified the generated and adapted code. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We are fully committed to reproducibility and have made every effort to provide the information needed to independently verify our theoretical results and reproduce our experiments. Appendix B states the assumptions underlying our theoretical contributions and provides detailed proofs. Appendix D presents practical implementation details and pseudocode for our proposed preference modelling and alignment methods. Appendix E documents the datasets, models, and experimental setup, including method-specific hyperparameter search ranges.

REFERENCES

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *International Conference on Learning Representations*, volume 2024, pp. 4965–4987, 2024.

Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

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

Fengyuan Dai, Zifeng Zhuang, Yufei Huang, Siteng Huang, Bangyan Liao, Donglin Wang, and Fajie Yuan. Vard: Efficient and dense fine-tuning for diffusion models with value-based rl. *arXiv preprint arXiv:2505.15791*, 2025.

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

Tobin A. Driscoll and Richard J. Braun. *Fundamentals of Numerical Computation*. Society for Industrial and Applied Mathematics, 2017. ISBN 978-1-61197-507-9. doi: 10.1137/1.9781611975086.

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, Chaoran Cheng, Shuaike Shen, Xiangxin Zhou, and Ge Liu. Fine-tuning flow matching generative models with intermediate feedback. *arXiv preprint arXiv:2510.18072*, 2025a.

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*, 2025b.

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.

Zheming Fu, Ruizhe He, Wei Shang, Xiaoxiao Ma, Lei Wang, Chang Liu, and Siming Fu. Flowawr: Online adaptive flow reinforcement via advantage-weighted rectification. *arXiv preprint arXiv:2606.30376*, 2026.

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

Dailan He, Guanlin Feng, Xingtong Ge, Yazhe Niu, Yi Zhang, Bingqi Ma, Guanglu Song, Yu Liu, and Hongsheng Li. Neighbor grpo: Contrastive ode policy optimization aligns flow models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6033–6042, 2026a.

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

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

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

Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *Advances in neural information processing systems*, 36:36652–36663, 2023.

Branislav Kveton, Anup Rao, Subhojyoti Mukherjee, Krishna Kumar Singh, and Viet Dac Lai. Advantageflow: Advantage-weighted least squares for rl in flow models. *arXiv preprint arXiv:2605.26013*, 2026.

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.

Niantong Li, Guangzheng Hu, Weixu Qiao, Ying Ba, Qichen Hong, Shijun Shen, Jinlin Wang, Fan Zhou, Jianye Kang, Xin Shang, et al. Qwen-image-bench: From generation to creation in text-to-image evaluation. *arXiv preprint arXiv:2605.28091*, 2026.

Youwei Liang, Junfeng He, Gang Li, Peizhao Li, Arseniy Klimovskiy, Nicholas Carolan, Jiao Sun, Jordi Pont-Tuset, Sarah Young, Feng Yang, et al. Rich human feedback for text-to-image generation. In *2024 ieee/cvf conference on computer vision and pattern recognition (cvpr)*, pp. 19401–19411. IEEE, 2024.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13199–13208. IEEE, 2025.

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.

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

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. *Advances in neural information processing systems*, 38:40783–40818, 2026a.

Jie Liu, Gongye Liu, Jiajun Liang, Ziyang Yuan, Xiaokun Liu, Mingwu Zheng, Xiele Wu, Qiulin Wang, Menghan Xia, Xintao Wang, et al. Improving video generation with human feedback. *Advances in Neural Information Processing Systems*, 38:82155–82192, 2026b.

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

Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, et al. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In *The twelfth international conference on learning representations*, 2023.

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

OpenAI. GPT-5.6: Frontier intelligence that scales with your ambition. https://openai.com/index/gpt-5-6/, July 2026. Published July 9, 2026; accessed August 31, 2026.

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *International conference on machine learning*, pp. 387–395. Pmlr, 2014.

Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022.

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

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

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

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.

Yuanming Yang, Guoqing Ma, Bo Wang, Yuan Zhang, Wei Tang, Chenyi Li, Haoyang Huang, and Nan Duan. Dit-reward: Generative representations for text-to-image reward modeling. *arXiv preprint arXiv:2606.23626*, 2026.

Tao Zhang, Cheng Da, Kun Ding, Huan Yang, Yan Li, Tingting Gao, Di Zhang, Shiming Xiang, Chunhong Pan, et al. Diffusion model as a noise-aware latent reward model for step-level preference optimization. *Advances in Neural Information Processing Systems*, 38:106202–106235, 2026.

Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su, Stefano Ermon, Jun Zhu, and Ming-Yu Liu. Diffusionnft: Online diffusion reinforcement with forward process. In *International Conference on Learning Representations*, volume 2026, pp. 134129–134150, 2026.

A Limitations and Future Work

Our work has two main limitations. First, BT-Flow is trained on synthetic state–velocity pairs derived from linear interpolations between sampled noise and offline chosen/rejected endpoints. These pairs may not fully reflect the states and velocities encountered during on-policy alignment, requiring the learned flow reward to generalize across this distribution mismatch. Second, Flow-VPA targets multi-step ($N \geq 2$) flow sampling under limited NFE budgets. Its applicability to one-step generation and shortcut models remains unexplored.

Future work includes addressing this distribution mismatch and extending flow-aware preference alignment to generative modalities beyond images. A particularly important direction is mitigating reward hacking, which emerged as a major bottleneck in our experiments. Developing more robust reward models and alignment methods could help translate further optimization into genuine improvements in human preference, rather than merely higher learned reward scores.

B Proofs

B.1 Proof of Proposition 4.1

Proposition 4.1 Restated. *Fix a prompt $\mathbf{c}$, an initial state $\boldsymbol{x}_0$, and a sampling budget $N \geq 2$. Let $\Delta t = 1/N$ and a uniform time grid $k = 0, \Delta t, \ldots, 1 - \Delta t$. Consider an Euler trajectory*

$$\tau_N = \left(\boldsymbol{x}_N^{(0)}, \boldsymbol{v}_N^{(0)}, \boldsymbol{x}_N^{(\Delta t)}, \boldsymbol{v}_N^{(\Delta t)}, \ldots, \boldsymbol{x}_N^{(1)}\right), \qquad \boldsymbol{x}_N^{(0)} = \boldsymbol{x}_0.$$

Construct a second Euler trajectory from the same initial state by adding a perturbation $\boldsymbol{\delta}_N^{(k)}$ to the realized velocity $\boldsymbol{v}_N^{(k)}$ at each step k. If

$$\sum_{k \in \{0, \Delta t, \ldots, 1-\Delta t\}} \boldsymbol{\delta}_N^{(k)} = \mathbf{0},$$

then the two trajectories receive the same image-level reward.

Proof. We first define the perturbed velocities from the original trajectory's realized velocities:

$$\tilde{\boldsymbol{v}}_N^{(k)} = \boldsymbol{v}_N^{(k)} + \boldsymbol{\delta}_N^{(k)}, \qquad k = 0, \Delta t, \ldots, 1 - \Delta t.$$

Starting from $\tilde{\boldsymbol{x}}_N^{(0)} = \boldsymbol{x}_0$, we construct the perturbed Euler trajectory $\tilde{\tau}_N$ recursively by

$$\tilde{\boldsymbol{x}}_N^{(k+\Delta t)} = \tilde{\boldsymbol{x}}_N^{(k)} + \Delta t\, \tilde{\boldsymbol{v}}_N^{(k)} = \tilde{\boldsymbol{x}}_N^{(k)} + \Delta t \left(\boldsymbol{v}_N^{(k)} + \boldsymbol{\delta}_N^{(k)}\right).$$

The first update therefore gives

$$\tilde{\boldsymbol{x}}_N^{(\Delta t)} = \boldsymbol{x}_0 + \Delta t \left(\boldsymbol{v}_N^{(0)} + \boldsymbol{\delta}_N^{(0)}\right).$$

Continuing this recursion, the state after i steps is

$$\tilde{\boldsymbol{x}}_N^{(i\Delta t)} = \boldsymbol{x}_0 + \Delta t \sum_{j=0}^{i-1} \left(\boldsymbol{v}_N^{(j\Delta t)} + \boldsymbol{\delta}_N^{(j\Delta t)}\right), \qquad i = 1, \ldots, N.$$

In particular, the perturbed endpoint is

$$\tilde{\boldsymbol{x}}_N^{(1)} = \boldsymbol{x}_0 + \Delta t \sum_{k \in \{0, \Delta t, \ldots, 1-\Delta t\}} \left(\boldsymbol{v}_N^{(k)} + \boldsymbol{\delta}_N^{(k)}\right).$$

Setting each perturbation as 0, the original endpoint is reconstructed as

$$\boldsymbol{x}_N^{(1)} = \boldsymbol{x}_0 + \Delta t \sum_{k \in \{0, \Delta t, \ldots, 1-\Delta t\}} \boldsymbol{v}_N^{(k)}.$$

Subtracting these expressions and applying the zero-sum condition yields

$$\tilde{\boldsymbol{x}}_N^{(1)} - \boldsymbol{x}_N^{(1)} = \Delta t \sum_{k \in \{0, \Delta t, \ldots, 1-\Delta t\}} \boldsymbol{\delta}_N^{(k)} = \mathbf{0}.$$

As Δt is strictly positive, the two terminal states coincide only if

$$\sum_{k\in\{0,\Delta t,\ldots,1-\Delta t\}} \boldsymbol{\delta}_N^{(k)} = \mathbf{0}.$$

Since the prompt $\boldsymbol{c}$ is also fixed, any image-level reward evaluated at the terminal state is identical for the two trajectories, which completes the proof.

B.2 Proof of Proposition 5.1

Proposition 5.1 Restated. *Suppose endpoint preferences follow the Bradley–Terry model in Eq. 1. Define the canonical latent flow reward*

$$r^\star_{\text{flow}}(\boldsymbol{x}, \boldsymbol{v}, t, \boldsymbol{c}) = r^\star(\boldsymbol{x} + (1-t)\boldsymbol{v}, \boldsymbol{c}),$$

and let $p^\star_{\text{flow}}$ denote its induced Bradley–Terry preference probability. Then, for every shared state $\boldsymbol{x}_t$, time $t < 1$, and endpoint pair $(\boldsymbol{x}_1^+, \boldsymbol{x}_1^-)$, the velocities constructed in Eq. 10 satisfy

$$p^\star_{\text{flow}}\left(\boldsymbol{v}_t^+ \succ \boldsymbol{v}_t^- \mid \boldsymbol{x}_t, t, \boldsymbol{c}\right) = p^\star\left(\boldsymbol{x}_1^+ \succ \boldsymbol{x}_1^- \mid \boldsymbol{c}\right).$$

Proof. Fix a shared state $\boldsymbol{x}_t$, time $t < 1$, prompt $\boldsymbol{c}$, and endpoint pair $(\boldsymbol{x}_1^+, \boldsymbol{x}_1^-)$. By Eq. 10, $\boldsymbol{v}_t^\pm = (\boldsymbol{x}_1^\pm - \boldsymbol{x}_t)/(1-t)$. Thus, for either sign,

$$r^\star_{\text{flow}}\left(\boldsymbol{x}_t, \boldsymbol{v}_t^\pm, t, \boldsymbol{c}\right) = r^\star\left(\boldsymbol{x}_t + (1-t)\frac{\boldsymbol{x}_1^\pm - \boldsymbol{x}_t}{1-t}, \boldsymbol{c}\right) = r^\star\left(\boldsymbol{x}_1^\pm, \boldsymbol{c}\right).$$

Substituting these identities into the Bradley–Terry model gives

$$\begin{aligned}
&p^\star_{\text{flow}}\left(\boldsymbol{v}_t^+ \succ \boldsymbol{v}_t^- \mid \boldsymbol{x}_t, t, \boldsymbol{c}\right) \\
&\quad= \sigma\left(r^\star_{\text{flow}}\left(\boldsymbol{x}_t, \boldsymbol{v}_t^+, t, \boldsymbol{c}\right) - r^\star_{\text{flow}}\left(\boldsymbol{x}_t, \boldsymbol{v}_t^-, t, \boldsymbol{c}\right)\right) \\
&\quad= \sigma\left(r^\star\left(\boldsymbol{x}_1^+, \boldsymbol{c}\right) - r^\star\left(\boldsymbol{x}_1^-, \boldsymbol{c}\right)\right) \\
&\quad= p^\star\left(\boldsymbol{x}_1^+ \succ \boldsymbol{x}_1^- \mid \boldsymbol{c}\right),
\end{aligned}$$

where the last equality follows from Eq. 1. This completes the proof.

B.3 Proof of Theorem 5.1

Theorem 5.1 Restated. Fix a source noise $\boldsymbol{x}_0$, a prompt $\boldsymbol{c}$, and $N \geq 2$ Euler steps. Suppose the latent reward $r^\star(\cdot, \boldsymbol{c})$ has a unique maximizer $\boldsymbol{x}_1^\star$, and the model can realize both rectified and non-rectified N-step Euler trajectories from $\boldsymbol{x}_0$ to $\boldsymbol{x}_1^\star$. Let $\Theta^\star_{\text{Image}}$ and $\Theta^\star_{\text{Flow}}$ denote the maximizer sets of $\mathcal{J}_{\text{ImageAlign}}(\theta; r^\star)$ and $\mathcal{J}_{\text{FlowAlign}}(\theta; r^\star_{\text{flow}})$ in Eqs. 9 and 13, respectively, evaluated for this fixed source–prompt pair on deterministic Euler rollouts with $\beta = 0$. Here $r^\star_{\text{flow}}$ is the canonical lift of $r^\star$ defined in Proposition 5.1. Let Θ_{dir} denote the set of parameters whose corresponding Euler rollouts are exactly direct rectified transports. Then

$$\Theta^\star_{\text{Flow}} = \Theta^\star_{\text{Image}} \cap \Theta_{\text{dir}} \subsetneq \Theta^\star_{\text{Image}}.$$

Proof. Write $\Delta t = 1/N$. For a parameter vector θ, let τ_N be its deterministic Euler rollout, with

$$\boldsymbol{v}_N^{(k)} = \boldsymbol{v}_\theta\left(\boldsymbol{x}_N^{(k)}, k, \boldsymbol{c}\right), \qquad \boldsymbol{x}_N^{(k+\Delta t)} = \boldsymbol{x}_N^{(k)} + \Delta t\, \boldsymbol{v}_N^{(k)}, \qquad \boldsymbol{x}_N^{(0)} = \boldsymbol{x}_0$$

for $k = 0, \Delta t, \ldots, 1 - \Delta t$. The rollout is direct precisely when

$$\boldsymbol{x}_N^{(k)} = (1-k)\boldsymbol{x}_0 + k\boldsymbol{x}_N^{(1)}, \qquad k = 0, \Delta t, \ldots, 1.$$

Let $r_{\max} = r^\star(\boldsymbol{x}_1^\star, \boldsymbol{c})$ and define the implied endpoint at each sampling time k by

$$\hat{\boldsymbol{x}}_{1,k} = \boldsymbol{x}_N^{(k)} + (1-k)\boldsymbol{v}_N^{(k)}.$$

Using the definition of $r^\star_{\text{flow}}$ and $\beta = 0$, the objectives for the fixed source–prompt pair become

$$\mathcal{J}_{\text{ImageAlign}}(\theta; r^*) = r^\star\left(\boldsymbol{x}_N^{(1)}, \boldsymbol{c}\right) \le r_{\max},$$
$$\mathcal{J}_{\text{FlowAlign}}(\theta; r^*_{\text{flow}}) = \frac{1}{N} \sum_{k\in\{0,\Delta t,\dots,1-\Delta t\}} r^\star\left(\hat{\boldsymbol{x}}_{1,k}, \boldsymbol{c}\right) \le r_{\max}.$$

On the assumed direct rollout to $\boldsymbol{x}_1^\star$, successive differences of the interpolation identity give $\boldsymbol{v}_N^{(k)} = \boldsymbol{x}_1^\star - \boldsymbol{x}_0$. Hence $\hat{\boldsymbol{x}}_{1,k} = \boldsymbol{x}_1^\star$ at every step, and both upper bounds are attained.

Since each summand is at most $r_{\max}$, the flow average equals $r_{\max}$ exactly when every summand does. The uniqueness of $\boldsymbol{x}_1^\star$ therefore gives

$$\theta \in \Theta^\star_{\text{Image}} \iff \boldsymbol{x}_N^{(1)} = \boldsymbol{x}_1^\star,$$
$$\theta \in \Theta^\star_{\text{Flow}} \iff \hat{\boldsymbol{x}}_{1,k} = \boldsymbol{x}_1^\star \quad \text{for every } k = 0, \Delta t, \dots, 1-\Delta t.$$

Now let $\theta \in \Theta^\star_{\text{Flow}}$. The second equivalence implies

$$\boldsymbol{v}_N^{(k)} = \frac{\boldsymbol{x}_1^\star - \boldsymbol{x}_N^{(k)}}{1-k}.$$

Starting from $\boldsymbol{x}_N^{(0)} = \boldsymbol{x}_0$, the Euler recursion yields

$$\boldsymbol{x}_N^{(k)} = (1-k)\boldsymbol{x}_0 + k\boldsymbol{x}_1^\star, \qquad k = 0, \Delta t, \dots, 1.$$

Indeed, whenever this identity holds at a sampling time $k < 1$, the displayed velocity equals $\boldsymbol{x}_1^\star - \boldsymbol{x}_0$, so the Euler update gives the same identity at $k + \Delta t$. Thus, by induction, the rollout is direct and reaches $\boldsymbol{x}_1^\star$. Consequently,

$$\Theta^\star_{\text{Flow}} \subseteq \Theta^\star_{\text{Image}} \cap \Theta_{\text{dir}}.$$

Conversely, any $\theta \in \Theta^\star_{\text{Image}} \cap \Theta_{\text{dir}}$ induces a direct rollout to $\boldsymbol{x}_1^\star$. As shown above, such a rollout satisfies $\hat{\boldsymbol{x}}_{1,k} = \boldsymbol{x}_1^\star$ at every step, so $\theta \in \Theta^\star_{\text{Flow}}$. This proves the set equality.

Finally, for the fixed source, endpoint, and time grid, the direct rectified trajectory is unique. By assumption, a non-direct trajectory to the preferred endpoint is realizable. Its realizing parameter belongs to $\Theta^\star_{\text{Image}}$ but not to Θ_{dir}, making the inclusion strict. This completes the proof.

C Extended Analysis of State-Based Preference Alignment

We extend our theoretical analysis to state-based preference alignment. We first describe reward learning from noised preference pairs and the resulting dense alignment objective. We then show that, although state rewards provide dense supervision to individual velocity predictions, their induced velocity preferences need not be consistent with direct rectified transport.

C.1 State-based Online Alignment

State-based reward models extend image-level preference modeling to intermediate states by conditioning their predictions on the noise level. Examples include the step-aware preference model in SPO (Liang et al., 2025) and the latent reward model LRM (Zhang et al., 2026). We formulate this noised reward-learning approach using the rectified-flow notation introduced in Section 3.

Given an offline preference pair $(\boldsymbol{c}, \boldsymbol{x}_1^+, \boldsymbol{x}_1^-) \sim \mathcal{D}$, we sample a time $t \sim (0, 1]$ and a shared source noise $\boldsymbol{x}_0 \sim p_0 = \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. The interpolation in Eq. 3 gives

$$\boldsymbol{x}_t^+ = (1-t)\boldsymbol{x}_0 + t\boldsymbol{x}_1^+, \qquad \boldsymbol{x}_t^- = (1-t)\boldsymbol{x}_0 + t\boldsymbol{x}_1^-.$$

Following the shared-noise construction used to train SPO's preference model (Liang et al., 2025), we reuse the endpoint preference label for this intermediate pair. This assumes that the preference ordering is preserved under corruption, although shared noise alone does not guarantee this consistency (Zhang et al., 2026). The resulting pairs train a time-conditioned reward model $r^{\text{state}}_\phi(\boldsymbol{x}, t, \boldsymbol{c})$ using the Bradley–Terry likelihood (Bradley & Terry, 1952) in Eq. 1:

$$\mathcal{L}^{\text{state}}_{\text{BT}}(\phi) = -\mathbb{E}_{(\boldsymbol{c},\boldsymbol{x}_1^+,\boldsymbol{x}_1^-)\sim\mathcal{D},\boldsymbol{x}_0\sim p_0,t\sim\mathcal{U}(0,1]}\left[\log\sigma\left(r^{\text{state}}_\phi(\boldsymbol{x}_t^+, t, \boldsymbol{c}) - r^{\text{state}}_\phi(\boldsymbol{x}_t^-, t, \boldsymbol{c})\right)\right]. \tag{15}$$

Once trained, the state reward model is held fixed during online alignment. To analyze its use as dense supervision, we replace the terminal reward in Eq. 9 with the average reward over the resulting states of all sampled transitions:

$$\mathcal{J}_{\text{StateAlign}}(\theta; r_\phi^{\text{state}}) = \mathbb{E}_{\boldsymbol{c},\boldsymbol{x}_0,\,\tau_N\sim P_\theta(\cdot|\boldsymbol{x}_0,\boldsymbol{c})}\left[\frac{1}{N}\sum_{k\in\{\Delta t,\ldots,1\}} r_\phi^{\text{state}}\left(\boldsymbol{x}_N^{(k)}, k, \boldsymbol{c}\right)\right] - \beta\mathbb{D}_{\text{KL}}\left[p_\theta||p_{\text{ref}}\right].$$

Each transition thus receives a reward based on its resulting state. Unlike image-based alignment, this objective provides dense, transition-specific supervision throughout sampling rather than a single terminal preference evaluation.

C.2 State Preference Alignment Is Also Flow-Blind

State-based alignment addresses the sparsity of endpoint evaluation, but dense supervision alone does not resolve flow-blindness. A next-state reward can distinguish candidate velocities through the states they produce. However, these induced preferences need not favor direct rectified transport to the preferred endpoint.

Training on straight interpolation paths does not close this gap. The Bradley–Terry loss in Eq. 15 supervises state preferences at individual times, but does not require the preferred states across time to form a direct rectified trajectory from each rollout's source noise. Under uniqueness, the state-reward maximizer is fixed for a given prompt and time, whereas the corresponding position on a direct trajectory also depends on the source noise. The following theorem shows that this mismatch can make non-direct transport globally preferable.

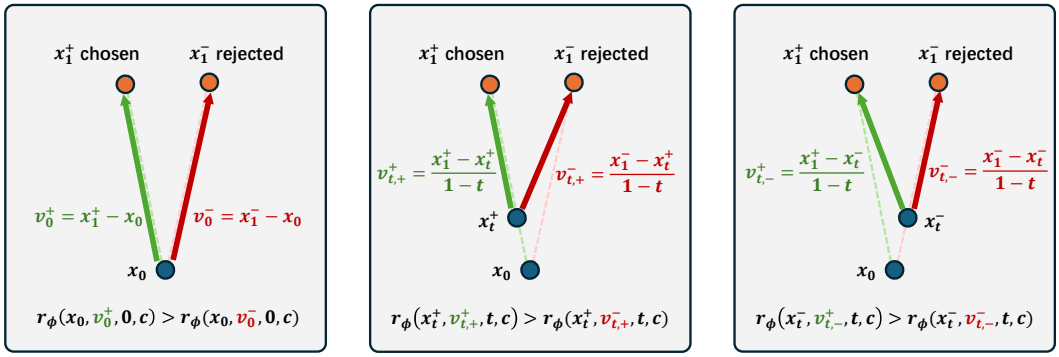
Proposition C.1 (Dense state rewards can favor non-direct transport) *Fix a prompt and deterministic $N \geq 2$-step Euler rollouts, consider ideal state alignment $\mathcal{J}_{\text{StateAlign}}(\theta; r^*_{\text{state}})$ separately for each source noise, using the latent r^*_{state} with $\beta = 0$. Suppose r^*_{state} has a unique maximizer at each sampled time, with terminal maximizer equal to the preferred endpoint $\boldsymbol{x}_1^\star$. From each source considered, assume that both direct rectified transport to $\boldsymbol{x}_1^\star$ and a trajectory visiting all statewise maximizers at their respective times are realizable. Then, except for at most one source noise, every globally optimal trajectory is non-direct and strictly outperforms the corresponding direct transport.*

Proof. Let $\boldsymbol{z}^\star$ be the state-reward maximizer at the first sampled time $\Delta t = 1/N < 1$. Direct transport through this state to $\boldsymbol{x}_1^\star$ requires

$$\boldsymbol{x}_0 = \frac{\boldsymbol{z}^\star - \Delta t\, \boldsymbol{x}_1^\star}{1 - \Delta t}.$$

Thus, the sequence of statewise maximizers can form direct transport from at most one source noise. For each source considered, a trajectory visiting all these maximizers attains the highest reward at every sampled time and is therefore globally optimal. By uniqueness, every global optimum must visit the same sequence of states. For any incompatible source, these optimal trajectories are non-direct. Direct transport instead misses at least one unique maximizer, receiving a strictly lower reward there and no higher reward elsewhere. Its cumulative objective is therefore strictly lower.

Dense supervision does not ensure direct transport. An endpoint reward leaves trajectories to the same image indistinguishable. A state reward can distinguish them, but may favor a non-direct route. The reason is structural: unique statewise maximizers specify intermediate destinations shared across source noises, whereas direct transport requires intermediate positions that depend on the source. Under the proposition's assumptions, this mismatch persists even with exact rewards and global optimization. Dense supervision can therefore resolve trajectory ambiguity without ensuring it in favor of rectified transport.



(a) Base case $t = 0$ (b) Chosen-endpoint anchor x_t^+ (c) Rejected-endpoint anchor x_t^-

Figure 4: **Dual-anchor construction of velocity preference pairs.** (a) At $t = 0$, the chosen and rejected endpoints define two direct-completion velocities from the shared source $\boldsymbol{x}_0$. (b, c) At an intermediate time $t < 1$, we construct two anchors by interpolating the same source toward each endpoint. At each anchor, we compare the velocities that complete the remaining transport to the chosen and rejected endpoints, holding the state, timestep, and prompt fixed. The endpoint preference supplies the label for both within-anchor comparisons. Arrows indicate the corresponding completion displacements.

D Practical Implementation

We discuss the practical design decisions for implementing BT-Flow and Flow-VPA. We first present the construction and representation of velocity preference pairs, followed by the tri-stream architecture with asymmetric attention. We then discuss Gaussian velocity exploration and its role in deterministic policy optimization. Figure 4 illustrates the dual-anchor pair construction, while Algorithm 1 summarizes the Flow-VPA alignment procedure.

D.1 BT-Flow Reward Model

Velocity pair construction. BT-Flow compares the preferred and rejected velocities defined in Eq. 10 under the same state, timestep, and prompt. During reward-model training, we sample

$$t \sim \mathrm{Uniform}(\{0, \frac{1}{N}, \ldots, \frac{N-1}{N}\}),$$

matching the timesteps used in the N-step policy-training rollouts.

Remaining-time rescaling. The direct-completion velocities $\boldsymbol{v}_t^\pm = (\boldsymbol{x}_1^\pm - \boldsymbol{x}_t)/(1-t)$ divide the endpoint residuals by the remaining transport time. Because the endpoints are distinct, a shared intermediate state cannot make both residuals vanish as t approaches 1. The factor $1/(1-t)$ can therefore amplify the candidate-velocity magnitudes near the endpoint.

To improve numerical conditioning, we rescale each candidate velocity by the remaining time before embedding it. For the training candidates, this exactly cancels the denominator:

$$(1-t)\boldsymbol{v}_t^\pm = \boldsymbol{x}_1^\pm - \boldsymbol{x}_t.$$

The reward model thus embeds the corresponding endpoint residuals. We implement this rescaling internally and retain the interface $r_\phi^{\text{flow}}(\boldsymbol{x}_t, \boldsymbol{v}_t, t, \boldsymbol{c})$ throughout the paper.

Dedicated velocity representation. The flow reward model architecture is adapted from the SD3.5-M backbone, following the MM-DiT design of separate visual and textual parameter streams. The candidate velocity shares the spatial structure of the visual latent but has semantics distinct from those of the visual and textual representations.

We therefore introduce a third stream dedicated to the candidate velocity. This stream receives the rescaled velocity and is initialized from the visual-stream parameters. A reward head reads hidden representations from selected layers of the velocity stream and outputs a scalar score.

Table 2: **Asymmetric attention for BT-Flow.** A check mark indicates that a row query may attend to the corresponding column key/value. Timestep conditioning uses the backbone's existing pathway and is omitted from the mask.

	Key/Value		
Query	$\boldsymbol{x}$	$\boldsymbol{c}$	$\boldsymbol{v}$
$\boldsymbol{x}$	✓	✓	–
$\boldsymbol{c}$	✓	✓	–
$\boldsymbol{v}$	✓	✓	✓

Candidate-independent conditioning. A dedicated velocity stream alone does not ensure the candidate-independent conditioning required by BT-Flow as in Eq. 11. Under fully bidirectional tri-stream with joint attention (Esser et al., 2024), candidate-velocity tokens could still modify the hidden visual and textual representations. The preferred and rejected velocities could therefore induce different internal conditioning contexts, despite sharing the same $(\boldsymbol{x}_t, t, \boldsymbol{c})$.

To preserve a shared internal conditioning context, we impose asymmetric attention: the visual and textual streams may attend to themselves and to each other, but not to the velocity stream, whereas the velocity stream may attend to all three streams. The velocity representation can thus incorporate the full conditioning context without altering the conditioning streams. Consequently, the reward head evaluates candidate-specific representations under the same internal condition. Table 2 summarizes the resulting attention pattern. The timestep enters through the backbone's existing conditioning pathway and is therefore omitted from the streamwise mask.

D.2 Deterministic Policy Optimization for Flow-VPA

For each policy update, we generate trajectories using the current deterministic sampler and construct a temporary batch $\mathcal{D}_\tau$ of state–time–prompt tuples $(\boldsymbol{x}, k, \boldsymbol{c})$ by randomly selecting a subset of the visited states from each trajectory. This subsampling reduces reward-evaluation and backpropagation costs without shortening the rollout. The selected states are held fixed during the update, and the batch is discarded afterward.

Stop-gradient state visitation. In Eq. 14, changing the policy affects both its velocity predictions and the state-visitation distribution d_N^θ. Flow-VPA differentiates through the predicted velocities while holding the visited states fixed within each update. $\mathcal{D}_\tau$ estimates expectations under $d_N^{\theta_i}$, but its sampled states are treated as constants during differentiation. Each batch is used for a single gradient update and then discarded. The visitation distribution therefore adapts between updates, without being differentiated through within an update.

This choice is motivated by the supervision that BT-Flow provides: at a fixed state, time, and prompt, it evaluates which velocity defines a more preferred rectified completion. In the canonical, unregularized setting of our theoretical analysis, the unique optimal endpoint $\boldsymbol{x}_1^\star$ determines the reward-maximizing velocity $\boldsymbol{v}^\star = (\boldsymbol{x}_1^\star - \boldsymbol{x})/(1-k)$. Realizing these local maximizers at every visited state produces direct rectified transport under Euler sampling. Thus, local velocity optimization is structurally consistent with the desired trajectory geometry, rather than relying solely on delayed endpoint feedback.

The update shares the sampled-state, action-gradient structure of deterministic actor methods (Silver et al., 2014; Lillicrap et al., 2015). The distinction is that Flow-RM evaluates local preferences, whereas an action-value critic accounts for subsequent rewards. Accordingly, our update optimizes a frozen-visitation surrogate, not an exact gradient of the full trajectory-dependent objective.

Gaussian velocity exploration. Evaluating the reward only at the current policy output provides a gradient at a single velocity. To probe nearby alternatives, we add independent Gaussian perturbations to the predicted velocity. This shares the action-level motivation of additive exploration noise in deterministic continuous-control methods such as DDPG (Lillicrap et al., 2015).

At each visited state, we draw independent noise samples and construct

$$\tilde{\boldsymbol{v}}_\theta(\boldsymbol{x}, k, \boldsymbol{c}) = \boldsymbol{v}_\theta(\boldsymbol{x}, k, \boldsymbol{c}) + \sigma_k \boldsymbol{\epsilon}_k, \qquad \boldsymbol{\epsilon}_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \boldsymbol{I}),$$

where σ_k is a velocity exploration scale. Unlike behavior-policy exploration noise in DDPG, these perturbations are used only for local reward evaluation: they are not propagated through the sampler and do not generate additional rollout states.

Motivated by the progression from noise-dominated states to endpoint refinement, we use the time-dependent exploration scale $\sigma_k = \sigma_{\max}(1-k)$. This schedule emphasizes exploration of alternative transport directions early and local velocity refinement later. The linear dependence on remaining transport time implements this design with a single scale hyperparameter, $\sigma_{\max}$.

Gaussian-smoothed reward estimation. Averaging the rewards of the perturbed velocities yields the empirical flow-reward term:

$$\tilde{\mathcal{J}}_{\text{FlowReward}}(\theta) = \mathbb{E}_{(\boldsymbol{x},k,\boldsymbol{c})\sim\mathcal{D}_\tau}\left[r_\phi^{\text{flow}}\left(\boldsymbol{x}, \tilde{\boldsymbol{v}}_\theta(\boldsymbol{x}, k, \boldsymbol{c}), k, \boldsymbol{c}\right)\right].$$

For a fixed rollout batch and policy, the effective reward expectation can be interpreted as the Gaussian-smoothed flow reward at the current policy output, where

$$\bar{r}_{\phi,\sigma}^{\text{flow}}(\boldsymbol{x}, \boldsymbol{v}, k, \boldsymbol{c}) = \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}\left[r_\phi^{\text{flow}}\left(\boldsymbol{x}, \boldsymbol{v} + \sigma_k\boldsymbol{\epsilon}, k, \boldsymbol{c}\right)\right].$$

Under this interpretation, σ_k determines the smoothing bandwidth. The decreasing noise scale therefore concentrates reward evaluation progressively closer to the current policy output.

Reference-regularized policy updates. We apply reference regularization to the unperturbed policy output, since the exploration perturbations are not part of the deployed velocity field:

$$\mathcal{J}_{\text{FlowRef}}(\theta) = \mathbb{E}_{(\boldsymbol{x},k,\boldsymbol{c})\sim\mathcal{D}_\tau}\left[\left\|\boldsymbol{v}_\theta(\boldsymbol{x}, k, \boldsymbol{c}) - \boldsymbol{v}_{\text{ref}}(\boldsymbol{x}, k, \boldsymbol{c})\right\|_2^2\right].$$

We maximize the resulting practical objective:

$$\begin{aligned}\hat{\mathcal{J}}_{\text{FlowVPA}}(\theta) &= \tilde{\mathcal{J}}_{\text{FlowReward}}(\theta) - \beta\mathcal{J}_{\text{FlowRef}}(\theta) \\ &= \mathbb{E}_{(\boldsymbol{x},k,\boldsymbol{c})\sim\mathcal{D}_\tau}\left[r_\phi^{\text{flow}}\left(\boldsymbol{x}, \tilde{\boldsymbol{v}}_\theta(\boldsymbol{x}, k, \boldsymbol{c}), k, \boldsymbol{c}\right) - \beta\left\|\boldsymbol{v}_\theta(\boldsymbol{x}, k, \boldsymbol{c}) - \boldsymbol{v}_{\text{ref}}(\boldsymbol{x}, k, \boldsymbol{c})\right\|_2^2\right]. \quad (16)\end{aligned}$$

We differentiate this objective with respect to θ, passing reward gradients through the perturbed velocities while holding the sampled states and noise fixed. The reward-model parameters ϕ and the reference policy remain frozen. The reward update uses the action-gradient structure of deterministic policy gradients (Silver et al., 2014), applied to BT-Flow's local reward rather than an action-value critic. No gradients propagate through the rollout transitions.

This local reward-gradient update differs from Flow-GRPO (Liu et al., 2026a), where an auxiliary SDE introduces stochastic transitions whose probability ratios enter the policy objective. Flow-VPA instead evaluates perturbed velocities at fixed on-policy states and updates the policy directly through flow-reward gradients. It thus decouples local velocity exploration from rollout dynamics: exploration noise shapes the learning signal rather than the sampling transitions.

E Experimental Details

E.1 Experimental Setup

Dataset and Model. We use Pick-a-Pic-v2 (Kirstain et al., 2023) and retain its existing training and evaluation splits. All reward and policy models are initialized from SD3.5-M (Esser et al., 2024). For reward training and offline alignment, we remove tied comparisons and examples with null prompts from the training split, yielding approximately 851.3K preference pairs. For online alignment, we discard the images and preference labels and deduplicate the training prompts, yielding approximately 58.9K unique prompts.

Reward Training. Image-RM and State-RM share the same backbone architecture and reward-head design. We extract features from DiT blocks 12, 16, 20, and 24, using image-token features

Algorithm 1 Flow-VPA with Velocity Exploration

Require: prompt dataset $\mathcal{D}_{\boldsymbol{c}}$, flow reward model r_ϕ^{flow}, reference velocity field $\boldsymbol{v}_{\text{ref}}$, deterministic rollout solver, rollout budget N, number of selected states $S \le N$, maximum exploration scale $\sigma_{\max}$, reference regularization coefficient β, and training iterations T.

Initialize θ_0 from the reference policy.
Freeze r_ϕ^{flow} and $\boldsymbol{v}_{\text{ref}}$.
Set $\Delta t = 1/N$ and the rollout time grid $k = 0, \Delta t, \dots, 1 - \Delta t$.
for $i = 0$ **to** $T - 1$ **do**
    Sample a prompt $\boldsymbol{c} \sim \mathcal{D}_{\boldsymbol{c}}$ and a source state $\boldsymbol{x}_N^{(0)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$.
    Using $\boldsymbol{v}_{\theta_i}$, generate the full N-step rollout τ_N as in Eq. 7, without gradient tracking.
    Randomly select a subset of rollout $\tau_{N,S}$, with $|\tau_{N,S}| = S$.
    Form the temporary on-policy state batch $\mathcal{D}_\tau$.
    for all $(\boldsymbol{x}, k, \boldsymbol{c}) \in \mathcal{D}_\tau$ **do**
        Sample independent Gaussian noise $\boldsymbol{\epsilon}_k \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$.
        Construct the perturbed velocity $\tilde{\boldsymbol{v}}_\theta(\boldsymbol{x}, k, \boldsymbol{c}) = \boldsymbol{v}_\theta(\boldsymbol{x}, k, \boldsymbol{c}) + \sigma_{\max}(1 - k)\boldsymbol{\epsilon}_k$.
    end for
    Update $\theta_i \to \theta_{i+1}$ to maximize $\hat{\mathcal{J}}_{\text{FlowVPA}}$ in Eq. 16 averaging over $\mathcal{D}_\tau$.
    Discard $\mathcal{D}_\tau$.
end for

for Image-RM and State-RM and velocity-token features for Flow-RM. These features are mean-pooled, and then apply classifier-free guidance (CFG) to the pooled representations, before passing to the reward head for a scalar score. Image-RM evaluates clean images at $t = 1$, whereas State-RM evaluates intermediate states at $t \in (0, 1]$.

We train full parameters of the DiT backbone and a scalar reward head using the Adam optimizer (Kingma & Ba, 2014), while keeping the VAE and all three text encoders frozen. All reward models use an image resolution of 512×512 and maximum sequence lengths of 256 for T5 and 77 for CLIP. For each reward formulation, we independently tune the learning rate over $\{1e{-}5,\ 3e{-}5,\ 5e{-}5,\ 1e{-}4\}$, the effective batch size over $\{64, 128, 256, 512\}$ image pairs, and the CFG scale over $\{1, 3, 4, 5, 6, 7\}$.

Text-to-Image Alignment. We fine-tune SD3.5-M using LoRA (Hu et al., 2021) with rank 32 and scaling parameter 64, and optimize the trainable parameters also with the Adam optimizer. Training consists of one complete epoch over the deduplicated prompt dataset, with 24 prompts per gradient update. For online methods, each rollout batch is generated by the current policy and used for a single gradient update, without off-policy buffer replay.

Training rollouts use 10 NFEs while retaining the solver and noise settings of the corresponding method implementation. All methods use a policy CFG scale of 4.5 and generate images at a resolution of 512×512, matching the resolution used to train the reward models. For clean and fair comparisons across all alignment methods, we disable exponential moving average (EMA) updates and use the initialization model as the fixed reference policy.

E.2 Flow-RM and Flow-VPA

We implement Flow-RM and Flow-VPA by adapting the training code and pipelines from Diffusers (von Platen et al., 2022), LRM (Zhang et al., 2026), and Flow-GRPO (Liu et al., 2026a).

We train Flow-RM for one epoch with an effective batch size of 128 image pairs, a learning rate of $5e{-}5$, and a CFG scale of 5. Then, we fine-tune the base model with Flow-VPA for one epoch. For each prompt, we initialize six rollouts, $K = 6$, from distinct source-noise samples. We use a learning rate of $3e{-}7$, a reference-regularization coefficient of $\beta = 0.1$, and a maximum exploration scale of $\sigma_{\max} = 0.5$. We randomly select five state–time pairs from each 10-step rollout for policy optimization, reducing reward-evaluation and backpropagation costs without shortening the rollout. We examine the effect of the number of selected states in Section E.6.

The hyperparameter search spaces for Flow-VPA and all baselines are summarized in Table 3.

Table 3: **Alignment hyperparameter search spaces.** Fixed method-specific parameters and settings are specified in the text.

Method	Hyperparameter Search Range
Flow-VPA	Learning Rate $\in$ [1e-7, 3e-7, 5e-7, 1e-6] $\beta \in$ [0.05, 0.1, 0.3, 0.5, 1.0] $\sigma_{\max} \in$ [0.1, 0.3, 0.5, 0.7]
Flow-DPO	Learning Rate $\in$ [3e-5, 5e-5, 1e-4, 3e-4] $\beta \in$ [3e2, 5e2, 1e3, 3e3, 5e3]
Flow-GRPO	Learning Rate $\in$ [3e-5, 5e-5, 1e-4, 3e-4] $\beta \in$ [0.05, 0.1, 0.3, 0.5, 1.0]
Flow-LPO	Learning Rate $\in$ [3e-5, 5e-5, 1e-4, 3e-4] $\beta \in$ [1e3, 3e3, 5e3, 1e4, 3e4] filter ratio $\in$ [0.25, 0.50, 0.75]

E.3 Baseline Methods

We reproduce the baselines using their official repositories or reimplement them with the adaptations described below. Beyond the shared experimental settings, we specify each method's rollout configuration and how its training signal is constructed. The method-specific hyperparameter search spaces are summarized in Table 3.

Flow-DPO. (Liu et al., 2026b) We consider Flow-DPO as an online Image-RM alignment baseline, we generate 24 images per prompt using the Euler ODE solver and rank them by their Image-RM scores. We construct a single preference pair per prompt from the highest-scoring and lowest-scoring images, designated as preferred and rejected, respectively.

Flow-GRPO. (Liu et al., 2026a) For online Image-RM alignment, we follow the reported rollout configuration, sampling 24 images per prompt using the auxiliary SDE with noise level 0.7. We compute group-relative advantages from their Image-RM endpoint scores. For online State-RM alignment, we adapt the method to compute advantages within state-local candidate groups. We initialize six rollouts per prompt and sample four candidate actions at each visited state. State-RM provides the scores used to compute group-relative advantages within each four-candidate group, and one candidate is selected at random to continue each rollout.

Flow-LPO. (Zhang et al., 2026) We adapt Latent Preference Optimization (LPO) to flow-model alignment using Flow-GRPO's auxiliary SDE with Euler discretization and noise level 0.7. We use the same sampling structure as the state-reward Flow-GRPO variant: six rollouts per prompt, four candidate actions per visited state, and one randomly selected candidate to continue each rollout. Rather than computing group-relative advantages, we construct a single local preference pair at each state, using the highest- and lowest-scoring candidates under State-RM as preferred and rejected, respectively. We retain LPO's confidence-based pair filtering and explicitly tune the filter ratio.

E.4 Evaluation Setup

We evaluate aligned models on the 500 prompts in the validation_unique split of Pick-a-Pic-v2, generating one image per prompt. All methods are evaluated on the same prompts at a resolution of 512×512 and with a CFG scale of 4.5, consistent with policy training. Unless otherwise specified, we use deterministic Euler ODE sampling with 40 NFEs. We assess the generated images using the complementary pairwise and pointwise metrics described below.

GPT-5.6-judged win rate. We use GPT-5.6 Luna and Terra to compare each aligned-model image against an image generated by the initialization model for the same prompt. We report the fraction of comparisons in which the aligned-model image is preferred. Unless otherwise specified, the reference images are generated by SD3.5-M using Euler sampling with 40 NFEs. For each comparison, we randomize the display order and require a forced choice between the two images. The judging prompt is provided in Appendix E.7.

Table 4: **External-evaluator agreement with decisive Pick-a-Pic-v2 human preferences.** Results use the validation_unique and test_unique splits. GPT agreement averages correctness over both display orders of each pair. Q-Judger agreement excludes invalid scores and exact ties, QAA-F averages Quality, Aesthetics, Alignment, and Fidelity; All5 additionally includes Creative Generation.

Evaluator	Validation-unique		Test-unique		API Cost ($)
	Agree (%)	95% CI	Agree (%)	95% CI	per 500 evals
GPT-5.6 Luna	63.52	[59.32, 67.60]	66.27	[62.38, 69.93]	0.175
GPT-5.6 Terra	64.34	[60.26, 68.41]	67.33	[63.44, 71.23]	1.749
GPT-5.6 Sol	65.62	[61.54, 69.70]	68.40	[64.39, 72.29]	3.446
Q-Judger					
Quality	51.95	[46.43, 57.47]	56.41	[50.96, 61.86]	—
Aesthetics	56.42	[50.68, 62.16]	56.72	[51.15, 62.30]	—
Alignment	61.54	[56.13, 66.67]	57.51	[52.31, 62.72]	—
Fidelity	52.13	[45.50, 58.77]	49.20	[41.71, 56.15]	—
QAA-F	59.14	[54.39, 63.90]	59.67	[54.89, 64.20]	—
All5	55.81	[50.65, 60.72]	60.85	[56.08, 65.61]	—

Q-Judger score. We additionally score each generated image individually using Q-Judger, the evaluator released with Qwen-Image-Bench (Li et al., 2026). We supply the corresponding Pick-a-Pic prompt and otherwise retain the configuration of the official implementation. We report the average of four pillars of Quality, Aesthetics, Alignment, and Fidelity as QAA-F. And we also report All5 which further includes Creative Generation as an additional dimension.

Agreement with human preferences. Separately from evaluating aligned-model outputs, we assess the external evaluators on the original human-labeled image pairs in the validation_unique and test_unique splits of Pick-a-Pic-v2. Table 4 reports agreement with decisive human preferences. For this assessment, GPT-5.6 judges each pair in both display orders, and correctness is averaged within each pair. For Q-Judger, we compare the two images' scores, excluding pairs with invalid or equal scores. These results show that both of the evaluators remain reasonable proxies for human preference evaluations.

E.5 Empirical Validation Setup

For the empirical analysis of flow-blindness, we examine the Flow-GRPO model aligned with Image-RM using the configuration selected through the training protocol and hyperparameter search described above. After alignment, we compare this model with its initialization on the 500 validation prompts using matched source-noise samples and NFE budgets $N \in \{10, 20, 40, 60, 80\}$. For each prompt and budget, both models start from the same source noise and are sampled using the first-order Euler solver. Unlike the default reference evaluation, this analysis uses the same NFE budget for the initialization and aligned models, while all other evaluation settings remain unchanged. We report the following three metrics.

Rectified-transport deviation. For a given Euler trajectory τ_N with a fixed $\Delta t = 1/N$, we measure deviation from direct rectified transport as

$$D_{\mathrm{rect}}(\tau_N) = \frac{1}{N-1} \sum_{k \in \{\Delta t, \ldots, 1-\Delta t\}} \frac{\left\| \boldsymbol{x}_N^{(k)} - \left[(1-k)\boldsymbol{x}_N^{(0)} + k\boldsymbol{x}_N^{(1)} \right] \right\|_2}{\left\| \boldsymbol{x}_N^{(1)} - \boldsymbol{x}_N^{(0)} \right\|_2}.$$

Each trajectory is compared with the straight-line interpolation between its own source and endpoint, normalized by the source-to-endpoint distance. Thus, $D_{\mathrm{rect}} = 0$ indicates direct rectified transport on the recorded grid, while $D_{\mathrm{rect}} = 0.05$ indicates an average deviation equal to 5% of the source-to-endpoint distance.

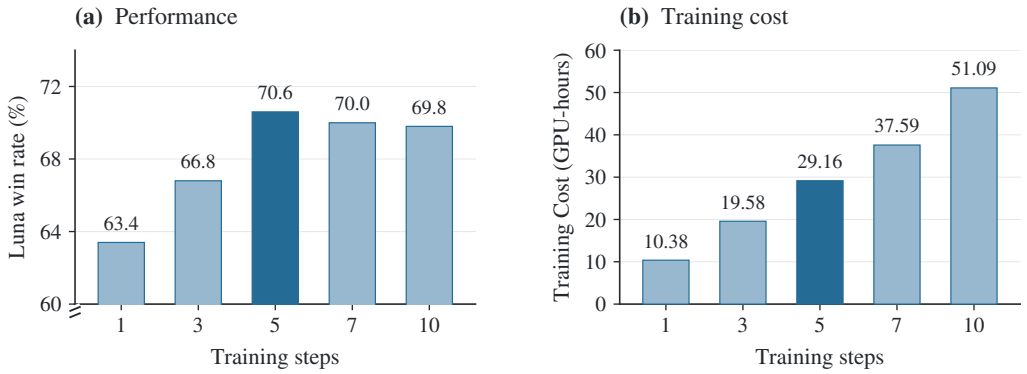


Figure 5: **State-subsampling ablation for Flow-VPA.** We vary the number of trained transitions per 10-step rollout and report (a) Luna preference win rate against the reference policy and (b) training cost in GPU-hours, calculated elapsed training hours multiplied by four for our four-H100 setup. We fix the learning rate to $3e{-}7$, $K = 1$, $\beta = 0.1$, and noise scale $\sigma_{\max} = 0.5$. Darker bars highlight five training steps, achieving a 70.6% win rate.

Image reward improvement. We measure the paired change in the Image-RM reward used for alignment:

$$\Delta r_N = r_\phi^{\text{img}}\left(\boldsymbol{x}_{N,\text{after}}^{(1)}, \boldsymbol{c}\right) - r_\phi^{\text{img}}\left(\boldsymbol{x}_{N,\text{before}}^{(1)}, \boldsymbol{c}\right),$$

where $\boldsymbol{c}$ is the validation prompt and the two endpoints are generated from the same source noise using the same NFE budget.

Pairwise preference win rate. We use GPT-5.6 Luna to compare the endpoint images before and after alignment for each matched prompt and source-noise sample. Following the evaluation protocol above, we randomize the display order and require a forced choice between the two images. We report the fraction of the comparisons in which the aligned-model image is preferred.

As shown in Figure 2, the aligned model exhibits greater rectified-transport deviation at higher NFE, while the difference becomes slightly negative at 10 NFE. Meanwhile, the endpoint reward improvement and pairwise win rate generally deteriorate with the decreasing NFE budget. Together, these observations show that preference improvement of endpoint image alignment can coexist with reduced trajectory rectification, while the resulting alignment gains are less fully realized at low NFE. This pattern is consistent with our flow-blindness analysis.

E.6 Ablation Study

Number of optimization states per rollout. We examine the performance–cost trade-off of state subsampling by fixing the rollout budget at $N = 10$ and varying the number of optimization states $S \in \{1, 3, 5, 7, 10\}$. This ablation uses one rollout per prompt ($K = 1$), with all other settings held fixed. As shown in Figure 5, training time grows approximately linearly with S, whereas alignment gains saturate. Increasing S from one to five improves the Luna win rate from 63.4% to 70.6%, but further increasing S to seven or ten yields no additional improvement (70.0% and 69.8%, respectively). Thus, optimizing more states continues to incur computational cost after the benefit of additional supervision has plateaued. Using five rather than all ten states reduces measured training time by 42.9% while achieving the highest observed win rate in this sweep. This trade-off motivates our use of $S = 5$ in the Flow-VPA experiments.

E.7 PAIRWISE IMAGE-PREFERENCE PROMPT

Table 5: **GPT-5.6 pairwise image-preference prompt.** The first three rows reproduce the system message, separated for readability. In the user message, c is the generation prompt, and I_A and I_B denote attached image inputs; the bracketed image placeholders are illustrative. The response schema restricts the `winner` field to `A` or `B`, with no tie option.

Component	Content
Task description	`You are predicting a human pairwise preference between two text-to-image generations produced for the same prompt.`
Preference criteria	`Choose the image that a human user is more likely to prefer overall. Jointly consider faithfulness to the prompt, correctness of depicted objects and their attributes or relationships, visible artifacts and image quality, composition, and aesthetic appeal.`
Decision rule	`Do not infer or reward the generating model's identity. Ignore presentation order. Even if the comparison is close, choose exactly one image: A or B.`
User message	`TEXT PROMPT:` $\{c\}$ `IMAGE A:` $\{I_A\}$ `IMAGE B:` $\{I_B\}$
Structured output	{`"winner": "A"`} or {`"winner": "B"`}