ADVANTAGE-GATED ANCHORING FOR DIFFUSION REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Advantage-weighted regression enables diffusion models to learn from reward feedback, providing a practical approach to diffusion reinforcement learning. For samples with positive advantages, regression encourages fitting their associated velocity targets. However, negative advantages push predictions away from these targets, without reallocating probability mass to alternative outcomes as negative feedback does in policies that explicitly predict action probabilities. This motivates studying policy anchoring as an additional constraint on velocity updates, acting as a reference for learning from negative feedback. In this work, we analyze how anchor policies shape reward learning and training stability in advantage-weighted regression and reveal that anchoring helps stabilize negative-advantage updates, while applying it to positive-advantage samples can hinder reward learning. Based on these insights, we propose **A**dvantage-**G**ated **A**nchoring (AGA), a simple yet effective method that anchors only negative-advantage samples. Experiments demonstrate faster reward improvement, a better balance between reward and image quality, and greater robustness to the choice of anchor policy.

1 INTRODUCTION

Reward-based fine-tuning helps image generators improve behaviors that are difficult to specify through pretraining alone, including accurate text rendering, compositional correctness, and alignment with human preferences (Black et al., 2024; Liu et al., 2025). Forward reinforcement learning uses this feedback by generating images, evaluating their rewards, and learning from their noised versions through the model's pretrained regression interface (Fan et al., 2025; Xue et al., 2026; Zheng et al., 2026). Advantage-weighted matching offers a direct way to perform this learning by weighting regression updates according to each sample's relative reward (Xue et al., 2026; Kveton et al., 2026; Choi et al., 2026a). Positive advantages encourage fitting the corresponding velocity targets, while negative advantages encourage moving predictions away from those targets.

However, negative feedback under this regression paradigm has a different meaning from that in conventional policies that explicitly predict action probabilities. Consider a discrete-action policy, such as a language model or robotic action model, in which actions are sampled from an explicit probability distribution obtained by applying softmax to logits (Achiam, 2018; Shao et al., 2024). Negative advantage penalizes the sampled action's log probability, and normalization ensures that reducing its probability increases the total probability mass assigned to alternative actions. In contrast, negative feedback in flow matching pushes predictions away from target velocities without redistributing probability mass to alternative outcomes. Besides, under an ELBO-based likelihood surrogate, increasing the matching error lowers a bound on log-likelihood but does not ensure that the sampled image becomes less likely (Black et al., 2024). Hence, the negative-advantage regression term can decrease without bound as prediction error grows, creating a source of training instability.

Policy anchoring provides a reference for these velocity updates by using an old policy's prediction at the same noisy input. Recent forward RL objectives incorporate such references through regularization and target corrections (Kveton et al., 2026; Fu et al., 2026; Zheng et al., 2026). We analyze their effects on reward learning and training stability through theory and experiments. Our analysis shows that anchoring can stabilize negative regression, but applying a shared rule to both feedback branches creates competing effects. Stronger anchoring retains old predictions at the cost of positive fitting, while reducing the shared baseline in an advantage-dependent rule can introduce repulsion from the

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.

anchor on positive samples. Besides, concentrating anchoring on negative samples improves the balance between reward learning and image quality and adding repulsion on positive samples instead produces rapid reward gains accompanied by severe quality loss and eventual training collapse.

These theoretical and empirical findings lead to *Advantage-Gated Anchoring* (AGA). AGA keeps ordinary regression for positive samples and evaluates the negative update at the old prediction. This replaces the negative quadratic with its local tangent approximation around an existing policy. Within the quadratic anchor family, it is the smallest nonnegative correction that removes negative curvature without adding further retention. The resulting rule has no independent anchor-strength parameter and removes the positive branch's direct dependence on the old prediction. AGA thus turns the analysis into a simpler way to use an existing policy anchor.

We evaluate AGA on compositional generation, text rendering, and human preference alignment within a shared training framework. AGA delivers strong compositional accuracy and improves text-rendering and preference rewards over the compared forward RL baselines under shared training configurations.

We make the following contributions.

- We analyze the role of policy anchoring in advantage-weighted flow matching, revealing the different anchoring needs of positive and negative feedback.
- We propose Advantage-Gated Anchoring (AGA), a simple rule that preserves positive fitting and anchors negative updates without an independent anchor-strength parameter.
- Through systematic experiments, we establish how anchor design shapes the reward–quality tradeoff and validate AGA's effectiveness across generation tasks and anchor-policy choices.

2 Related Work

Reward and preference optimization for diffusion models. Diffusion alignment incorporates feedback through both sampling trajectories and regression objectives. DDPO applies policy gradients to stochastic denoising transitions, DPOK adds KL regularization, and Flow-GRPO trains flow models with group-relative rewards (Black et al., 2024; Fan et al., 2023; Liu et al., 2025). With differentiable rewards, ReFL and DRaFT backpropagate through denoising computations (Xu et al., 2023; Clark et al., 2024). Diffusion-DPO learns from pairwise preferences through an ELBO-based objective, while Diffusion-KTO uses per-image binary feedback (Wallace et al., 2024; Li et al., 2024). Forward RL incorporates reward feedback by learning from noised versions of generated images. AWM derives advantage-weighted matching estimators and clipped ratio objectives (Xue et al., 2026), while FPO constructs PPO-style ratios from flow-matching losses (McAllister et al., 2026). RAM connects regression updates to adjoint matching (Bergmeister et al., 2026). EPG and PEPG examine the role of likelihood estimation, and path-space analysis further relates trajectory and forward policy-gradient estimators (Choi et al., 2026b; Xu et al., 2026). We build on this regression interface to study how policy anchors shape learning from positive and negative advantages.

Policy anchoring and asymmetric feedback. Reference predictions can constrain model drift and modify how reward feedback enters regression. AdvantageFlow analyzes negative-weight curvature and compares fixed and adaptive regularization (Kveton et al., 2026). FlowAWR constructs old-centered rectification targets, while RVM organizes velocity objectives into a family with a configurable reference coefficient (Fu et al., 2026; Choi et al., 2026a). A related line of work treats positive and negative feedback differently. Negative-Aware Fine-Tuning preserves positive fitting and constructs an implicit negative policy for language models, while DiffusionNFT uses implicit positive and negative forward policies for diffusion models (Chen et al., 2026; Zheng et al., 2026). CreFlow studies reflected negative targets and uses successful samples to construct corrections (Ni et al., 2026). The choice of reference also varies across methods. Centered Reward Distillation separates sampling and moving-reference policies, while DiffusionCompass incorporates high-quality off-policy examples as quality anchors (Zhu et al., 2026; Liu et al., 2026). Within this line of work, we examine where anchoring helps or obstructs advantage-weighted learning, connecting its effects on each feedback branch to reward, image quality, and robustness across anchor policies. These insights lead to AGA, which preserves positive fitting and uses the anchor only for negative updates.

3 UNDERSTANDING AND DESIGNING POLICY ANCHORS

Forward RL uses velocity regression to learn from rewards on generated images. We first examine how this interface represents positive and negative feedback, then study anchoring as a correction to the resulting updates. The distinction between negative-error maximization and positive fitting guides the analysis, method, and experimental comparisons.

3.1 PRELIMINARIES

Flow matching. For a prompt c, a rollout policy generates $x_0 \sim \pi_{\mathrm{b}}(\cdot \mid c)$. Conditional flow matching learns a velocity field along a path between images and Gaussian noise (Lipman et al., 2023). For the linear path $x_t = (1-t)x_0 + t\epsilon$, with $t \in [0,1]$ and $\epsilon \sim \mathcal{N}(0, I)$, the target velocity is $u = \epsilon - x_0$ and the model predicts $v = v_\theta(x_t, t, c)$. The ordinary regression objective is $\mathcal{L}_{\mathrm{FM}} = \frac{1}{2}\mathbb{E}\|v-u\|^2$. Its expectation covers prompts, generated images, sampled times, and noise. Generation integrates the learned field from noise at $t=1$ toward an image at $t=0$. Forward RL reuses this regression interface to learn from rewards on the generated images.

GRPO. Following GRPO (Shao et al., 2024), feedback is computed by comparing rewards within a group of images generated for the same prompt. For rewards $r_1, \ldots, r_G$, a standard group-relative advantage and its bounded form are

$$A_i = \frac{r_i - \mathrm{mean}_j(r_j)}{\mathrm{std}_j(r_j) + \varepsilon_{\mathrm{adv}}}, \qquad a_i = \frac{\mathrm{clip}(A_i, A_{\min}, A_{\max})}{A_{\max}}, \qquad A_{\min} = -A_{\max} < 0.$$

Here $\varepsilon_{\mathrm{adv}} > 0$ stabilizes normalization. Clipping and rescaling give $a_i \in [-1, 1]$ while preserving the sign of the feedback. We use a for a general bounded advantage below.

Diffusion RL. The advantages above provide feedback for adapting the generation policy. At the sampling policy, the loss that maximizes a likelihood-ratio objective has single-sample gradient $-a\nabla_\theta \log p_\theta(x_0 \mid c)$. For diffusion models, however, evaluating image likelihood requires marginalizing latent denoising trajectories. The ELBO supplies a tractable connection between this likelihood and the matching objectives used to train the model (Ho et al., 2020; Black et al., 2024). Let $q(x_{1:T} \mid x_0, c)$ be a fixed forward noising process and $p_\theta(x_{0:T} \mid c)$ the generative trajectory distribution. Marginalizing the latent trajectory and applying Jensen's inequality gives

$$\log p_\theta(x_0 \mid c) = \log \mathbb{E}_q\left[\frac{p_\theta(x_{0:T} \mid c)}{q(x_{1:T} \mid x_0, c)}\right] \geq \mathbb{E}_q\left[\log \frac{p_\theta(x_{0:T} \mid c)}{q(x_{1:T} \mid x_0, c)}\right] = \mathrm{ELBO}_\theta(x_0, c).$$

Thus $-\mathrm{ELBO}_\theta$ upper-bounds the negative image log-likelihood. With fixed Gaussian reverse variances, its trainable transition terms reduce to weighted squared denoising errors (Ho et al., 2020). For the linear flow-matching path above, the noise prediction induced by the velocity is $\epsilon_\theta = x_t + (1-t)v$, giving

$$\epsilon_\theta - \epsilon = (1-t)(v-u), \qquad \|\epsilon_\theta - \epsilon\|^2 = (1-t)^2\|v-u\|^2.$$

This transfers denoising error to velocity error, with the corresponding time-dependent weight. Previous work uses this matching interface for reward-based fine-tuning (Xue et al., 2026; Zheng et al., 2026; Kveton et al., 2026). Substituting negative matching error for the log-likelihood score in the advantage-weighted update leads to the reward-regression surrogate

$$\mathcal{L}_{\mathrm{adapt}} = \mathbb{E}[\ell_a(v)], \qquad \ell_a(v) = \frac{a}{2}\|v-u\|^2. \tag{1}$$

The advantage is fixed during differentiation. Noise-state weights and active importance ratios can scale this reward gradient, while policy regularization is specified separately. Appendix A derives the surrogate-gradient connection. Equation 1 is our starting reward objective. It fits positive-advantage targets and increases error on negative-advantage targets.

3.2 MOTIVATION: ASYMMETRIC FEEDBACK IN FLOW MATCHING

Advantage weighting places successful and unsuccessful samples in the same regression objective, but reversing the weight does not produce symmetric supervision. We examine this asymmetry by

comparing flow matching with policies that explicitly predict action probabilities. Both interfaces reinforce successful samples under positive advantage. Under negative advantage, suppressing an explicit action probability redistributes mass to alternatives, whereas increasing velocity error does not specify which generated outcomes should become less likely. Figure 1 illustrates these different effects.

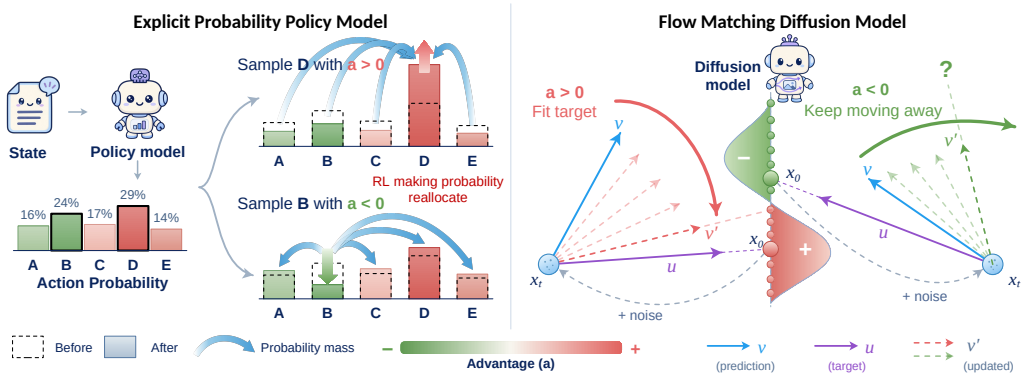

Figure 1: **Positive and negative feedback across prediction interfaces.** With the input and advantage fixed, positive feedback reinforces an action or fits a velocity target. Negative feedback redistributes explicit probabilities but drives velocity predictions away from the target. For an intuitive view of denoising, all velocity arrows are shown with reversed directions.

Feedback through explicit action probabilities. Discrete-action agents and language models commonly predict candidate action probabilities explicitly, normalizing their logits with softmax (Mnih et al., 2016; Shao et al., 2024). At a state s_t, e.g., a language-model prefix, the logit h_j gives $\pi(j \mid s_t) = \exp(h_j)/\sum_i \exp(h_i)$. For a sampled action $a_t = k$, write $p_k = \pi(a_t \mid s_t)$. Its advantage-weighted reward objective is $a \log p_k$ (Achiam, 2018), whose derivative with respect to h_k is $a(1 - p_k)$. Thus positive advantage raises the selected logit, while negative advantage lowers it. Holding the other logits fixed, softmax normalization gives

$$\frac{\partial p_k}{\partial h_k} = p_k(1 - p_k) > 0, \qquad \frac{\partial p_j}{\partial h_k} = -p_j p_k < 0 \quad (j \neq k).$$

As the probability bars on the left of Figure 1 show, raising h_k increases the selected probability, while lowering it reduces that probability and redistributes the removed mass to other actions according to their relative probabilities.

Feedback through velocity error. Flow matching instead uses the update $-\nabla_v \ell_a = a(u - v)$. As the right side of Figure 1 shows, positive advantage moves the prediction toward u, where the update vanishes. Under the ELBO construction in Section 3.1, fitting the appropriately weighted velocity targets raises a lower bound on image log-likelihood, providing a surrogate for reinforcing rewarded images.

Negative advantage reverses the arrow and pushes the prediction away from u. Increasing matching error lowers the likelihood bound, but does not ensure that the image itself becomes less likely (Black et al., 2024). Unlike the probability redistribution on the left side of Figure 1, this update changes the generation field without explicitly transferring probability mass to alternative outcomes. Figure 1 illustrates this distinction with a central curve showing rollout samples' advantages and dashed arrows showing successive predictions v' at each fixed noisy input. Positive feedback draws these predictions toward u, whereas negative feedback lacks this attracting endpoint and can continue driving them away, destabilizing training.

3.3 Policy Anchoring

To constrain these negative updates, diffusion RL can introduce an anchor policy whose predictions provide a reference for the reward update. Let $v_{\text{anc}} = v_{\theta_{\text{anc}}}(x_t, t, c)$ be the detached old-policy

prediction at the same noisy input. Adding a quadratic anchor to reward regression and taking the negative gradient gives

$$\ell_\omega(v; v_{\text{anc}}) = \ell_a(v) + \frac{\omega}{2}\|v - v_{\text{anc}}\|^2, \qquad F_\omega(v; v_{\text{anc}}) = -\nabla_v \ell_\omega = a(u - v) + \omega(v_{\text{anc}} - v). \quad (2)$$

Here F_ω is the prediction-space update direction, with $a, u, v_{\text{anc}}, \omega$ fixed during differentiation. Positive ω attracts the current prediction to v_{anc}, while negative ω repels it. In practice, the anchor policy is typically maintained as an exponential moving average (EMA) of the trainable model's parameters.

Many recent forward RL objectives can be expressed in this reward-and-anchor gradient form after identifying their effective matching coefficients and nonnegative sample scales (Xue et al., 2026; Kveton et al., 2026; Fu et al., 2026; Zheng et al., 2026). Appendix C derives these equivalences. With fixed or advantage-adaptive anchoring, the corresponding objective is

$$\ell_\omega(v; v_{\text{anc}}) = \frac{a}{2}\|v - u\|^2 + \frac{\omega(a)}{2}\|v - v_{\text{anc}}\|^2, \qquad \omega(a) = \beta - \gamma a. \quad (3)$$

Here $\beta, \gamma \geq 0$. The baseline β sets a shared anchoring strength. Like a fixed-reference KL penalty, it regularizes departure from a policy reference, here through the old velocity prediction. The parameter γ adjusts this strength with advantage, increasing anchoring on negative samples and reducing it on positive samples. Using one shared rule for both branches creates two problems.

Strong anchoring restricts positive fitting. Anchoring used to stabilize negative regression also retains the old prediction on positive samples. For $a > 0$ and $\omega > 0$, the anchored objective is minimized at $v^\star = (au + \omega v_{\text{anc}})/(a + \omega)$, so the prediction is driven toward a point between v_{anc} and u rather than the rewarded target u itself. Increasing the anchor coefficient shifts this solution toward v_{anc}, restricting adaptation to u. This extra retention can limit positive learning even when training already includes a fixed-reference KL penalty. A strong shared anchor therefore constrains both the negative updates it is intended to control and the positive fitting that already has a target.

Weak anchoring can leave negative regression uncorrected or repel positive predictions. For a negative sample, the reward quadratic has curvature aI and the anchor adds ωI. A fixed coefficient below $|a|$ leaves negative curvature $(a + \omega)I$, so reducing shared retention can fail to correct negative regression. An adaptive rule with $\gamma = 1$ instead leaves curvature βI, but lowering its baseline can change positive anchoring into repulsion. When $a > \beta$, its coefficient $\omega = \beta - a$ is negative. For $\beta > 0$, the anchored minimum is $v_{\text{anc}} + (a/\beta)(u - v_{\text{anc}})$, lying beyond u rather than simply releasing the fitting constraint. At $\beta = 0$, the rule $\omega = -a$ leaves the update $a(u - v_{\text{anc}})$, which need not vanish when the positive target has been fitted. Weakening a shared rule can thus remove the correction to negative regression or introduce an additional objective of moving positive predictions away from the anchor.

Section 4.3 tests these effects by varying anchor strength and its application to positive and negative samples. We track reward improvement together with image quality to examine how anchoring affects learning and training stability.

3.4 Advantage-Gated Anchoring

The anchor should correct negative regression without retaining or repelling predictions whose rewarded targets can already be fitted. A shared baseline ties these distinct requirements together. We therefore separate the two advantage branches, preserving ordinary positive fitting and applying only the correction needed by negative regression.

For $a \geq 0$, we set $\omega = 0$ to retain the original fitting target. For $a < 0$, we use the smallest nonnegative anchor coefficient that removes the reward term's negative curvature, namely $\omega = -a$. The resulting negative objective is a local linear approximation,

$$\ell_a(v) + \frac{-a}{2}\|v - v_{\text{anc}}\|^2 = \ell_a(v_{\text{anc}}) + a\langle v_{\text{anc}} - u, v - v_{\text{anc}}\rangle \;\geq\; \ell_a(v), \qquad a < 0. \quad (4)$$

It is the tangent upper bound of the concave reward term at v_{anc} (Yuille & Rangarajan, 2003; Mairal, 2015). Thus the anchor replaces negative quadratic regression with a local linear approximation

around the old prediction, preserving the reward gradient at that reference. Negative feedback uses the discrepancy between the old prediction and the sampled target to define the update. A larger coefficient would add retention around v_{anc} beyond this local correction.

Combining this negative correction with unchanged positive fitting gives *Advantage-Gated Anchoring* (AGA),

$$\omega_{\text{AGA}}(a) = -a\,\mathbf{1}[a<0], \qquad F_{\text{AGA}}(v; v_{\text{anc}}) = \begin{cases} a(u-v), & a \geq 0, \\ a(u-v_{\text{anc}}), & a < 0. \end{cases} \tag{5}$$

Here $\mathbf{1}[\cdot]$ denotes the indicator function. The piecewise field makes the branch-specific correction explicit. Positive samples retain $a(u-v)$, which vanishes at their fitting target. Negative samples use $a(u-v_{\text{anc}})$, evaluating their reward direction at the old prediction rather than increasing repulsion as v departs from u. Zero advantage contributes no reward update, and the coefficient follows from the local correction without an independent anchor-strength parameter. Algorithm 1 summarizes training, with sg denoting stop-gradient and $\mathcal{L}_{\text{ref}}$ a fixed-reference penalty.

Algorithm 1 Advantage-Gated Anchoring (AGA).

Require: Model v_θ, its policy copies, and reward function r.

for each training iteration **do**
  Sample $x_0 \sim \pi_{\text{b}}(\cdot \mid c)$ in prompt groups and compute advantages a.
  $x_t \leftarrow (1-t)x_0 + t\epsilon, \quad u \leftarrow \epsilon - x_0$ for sampled t, ϵ.
  $v \leftarrow v_\theta(x_t, t, c), \qquad v_{\text{anc}} \leftarrow \text{sg}[v_{\theta_{\text{anc}}}(x_t, t, c)]$
  $\omega \leftarrow \max(-a, 0)$
  Update θ using $\mathbb{E}[\frac{a}{2}\|v-u\|^2 + \frac{\omega}{2}\|v-v_{\text{anc}}\|^2] + \mathcal{L}_{\text{ref}}$.
  Update the policy copies by EMA.
end for

4 Experiments

4.1 Experimental setup

We evaluate SD3.5-Medium (Esser et al., 2024; Stability AI, 2024) on GenEval, OCR, and PickScore for compositional generation, text rendering, and preference alignment (Ghosh et al., 2023; Kirstain et al., 2023). We report task rewards on held-out prompts and image quality on DrawBench using ImageReward, aesthetic, CLIP, and PickScore (Saharia et al., 2022; Xu et al., 2023). For a fair comparison of all baselines, we use a unified trainer and keep its hyperparameters fixed within each comparison, varying only the learning objective, as detailed in Appendix D.

4.2 Performance

Figure 2 compares reward trajectories across three tasks. AGA and AWM rapidly reach comparable high GenEval rewards, while DiffusionNFT plateaus earlier and AdvantageFlow improves more slowly. On OCR, AGA continues improving after DiffusionNFT begins losing reward, and AWM learns more slowly. RAM initially improves rapidly but subsequently loses most of its OCR reward. AdvantageFlow's OCR reward gradually declines.

On PickScore, AGA continues improving after AWM, DiffusionNFT, and AdvantageFlow approach plateaus, while RAM's early gains give way to a sharp decline. These trajectories show strong compositional performance and more sustained reward gains on text rendering and preference alignment. Table 1 reports checkpoint rewards and DrawBench image-quality scores.

4.3 Analysis and Ablation

Following the analysis in Section 3.3, we further examine the effects of policy anchoring on diffusion RL. We conduct the coefficient experiments on the OCR task, tracking OCR reward alongside image quality measured by DrawBench PickScore to reveal reward improvement and reward hacking, where reward gains accompany degraded image quality.

Model	step	Task reward			DrawBench			
		GenEval	OCR	PickScore	ImgRwd	Aesthetic	CLIP	PickScore
SD3.5M w/o CFG		0.28	0.13	19.94	-0.50	5.16	0.2379	20.74
SD3.5M w/ CFG		0.63	0.57	21.78	0.92	5.44	0.2868	22.45
Compositional Image Generation								
AWM	200	**0.94**			**0.75**	5.30	0.2791	**22.10**
DiffusionNFT	200	0.82			0.48	5.19	0.2749	21.78
AdvantageFlow	200	0.79			0.41	5.27	0.2706	21.86
RAM	200	0.93			0.56	**5.33**	**0.2795**	21.81
AGA	200	**0.94**			0.54	5.15	0.2752	21.91
Visual Text Rendering								
AWM	300		0.79		0.80	5.27	0.2875	21.93
DiffusionNFT	300		0.82		**0.83**	5.33	**0.2891**	22.17
AdvantageFlow	230		0.45		0.77	5.44	0.2804	22.17
RAM†	150		0.93		0.74	**5.50**	0.2832	**22.30**
AGA	300		**0.96**		**0.83**	5.31	0.2861	22.23
Human Preference Alignment								
AWM	400			22.69	1.02	5.97	0.2638	23.07
DiffusionNFT	400			22.38	0.96	5.89	0.2659	22.78
AdvantageFlow	400			22.27	0.91	5.89	0.2610	22.64
RAM†	120			22.85	1.24	5.74	**0.2854**	23.21
AGA	400			**23.52**	**1.29**	**6.30**	0.2707	**23.89**

Table 1: **Reward and image quality on SD3.5M.** Best values within each task are bold and second-best values are underlined. † marks runs that subsequently collapse, reported at a pre-collapse step.

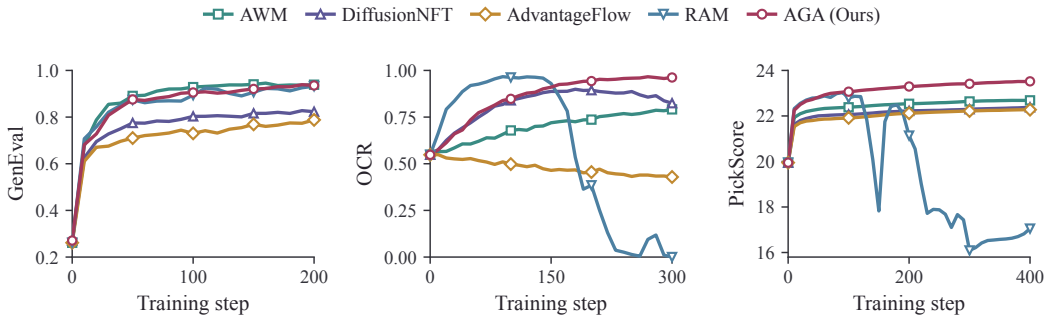

Figure 2: **Reward convergence across three tasks.** Each panel tracks held-out reward under a shared training configuration.

Moderate anchoring suppresses reward hacking without blocking learning. We first isolate the contribution of the anchor by comparing constant coefficients $\omega \in \{0, 0.1, 1\}$. Figure 3 shows that without anchoring, OCR initially improves rapidly while PickScore deteriorates, followed by a collapse in OCR. A modest coefficient of 0.1 instead sustains reward improvement with much smaller quality loss. With $\omega = 1$, OCR instead gradually declines, without a PickScore advantage over $\omega = 0.1$. Anchoring can therefore suppress the reward–quality divergence that precedes collapse, but excessive retention obstructs reward acquisition. This tradeoff motivates examining where the constraint is most useful.

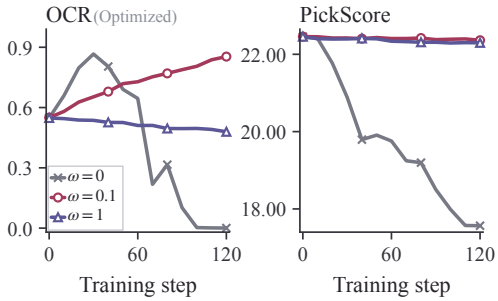

Figure 3: **The benefit and cost of anchoring.** Moderate anchoring retains image quality while improving OCR. A stronger coefficient prevents reward gains.

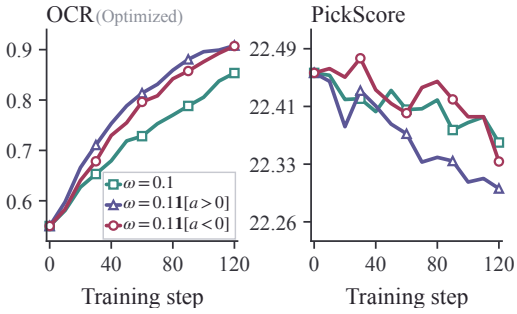

Figure 4: **Negative-sample anchoring is more effective.** It learns faster than all-sample anchoring and approaches positive-only OCR with better PickScore.

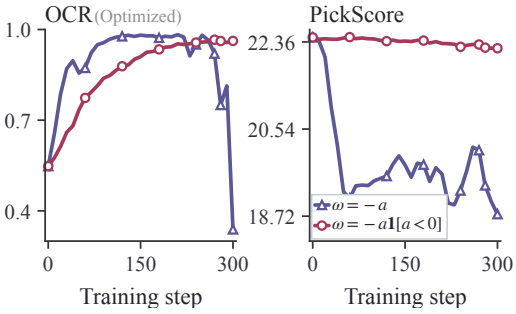

Figure 5: **Positive-sample repulsion.** $\omega = -a$ raises OCR while PickScore deteriorates, before OCR collapses.

Anchoring negative samples uses the constraint more effectively. The two feedback branches place different demands on an anchor. Negative regression needs a reference correction, while positive regression already has a target to fit. We compare $\omega = 0.1$, $0.1 \cdot \mathbf{1}[a > 0]$, and $0.1 \cdot \mathbf{1}[a < 0]$ to locate where the same active coefficient is most useful. In Figure 4, removing positive-sample anchoring speeds OCR learning relative to anchoring all samples, while largely retaining its PickScore trajectory. Removing negative-sample anchoring also accelerates OCR, and is initially faster, but sacrifices more PickScore. The two gated rules approach the same OCR level by the end of the comparison, with negative-only anchoring retaining higher quality. Anchoring negative samples therefore provides a more effective balance between reward acquisition and quality preservation, without constraining the positive fitting direction.

Negative anchor coefficients on positive samples aggravate reward hacking. Releasing positive fitting removes its competition with the anchor. Making the coefficient negative goes further by rewarding departure from the old prediction. We isolate this change by comparing $\omega = -a$ with AGA's $\omega = -a\,\mathbf{1}[a < 0]$, which keeps the negative branch identical. On positive samples, the former replaces ordinary fitting $a(u - v)$ with $a(u - v_{\text{anc}})$, removing its attenuation as v approaches u. Figure 5 exhibits a pronounced reward-hacking pattern. OCR rises faster and remains high over much of training even as DrawBench PickScore drops sharply and stays well below AGA. The early reward advantage thus conceals a substantial loss of general image quality. OCR subsequently collapses, whereas AGA continues toward a stable high reward with much better PickScore. Preserving ordinary positive fitting avoids the additional anchor repulsion that drives this reward–quality divergence.

AGA is more efficient than increasing the shared anchor strength. The preceding comparisons support preserving positive fitting while correcting negative updates. Increasing the shared baseline β also strengthens the constraint on negative updates, but extends it to positive samples. We compare AGA with $\omega = 1$ and $\omega = 1 - a$, two shared rules with $\beta = 1$ studied in previous work (Kveton et al., 2026). Both assign at least as much anchoring as AGA across $a \in [-1, 1]$, while also retaining positive predictions.

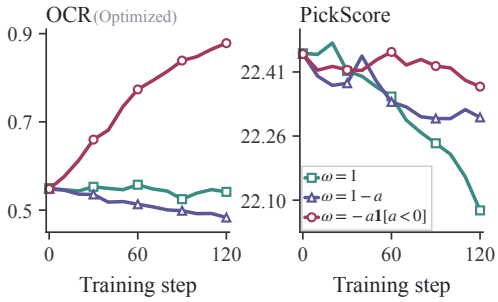

Figure 6: **AGA is more efficient than stronger shared anchoring.** AGA improves OCR faster and retains higher late-window PickScore than the two rules with $\beta = 1$.

Under the same advantage processing, sample weighting, and policy updates, Figure 6 shows that AGA improves OCR rapidly while maintaining PickScore. The fixed rule makes little OCR progress, and the adaptive rule loses OCR reward. Neither stronger rule provides a corresponding quality advantage, and both have lower PickScore later in the comparison. These results show that AGA achieves a better reward–quality tradeoff than

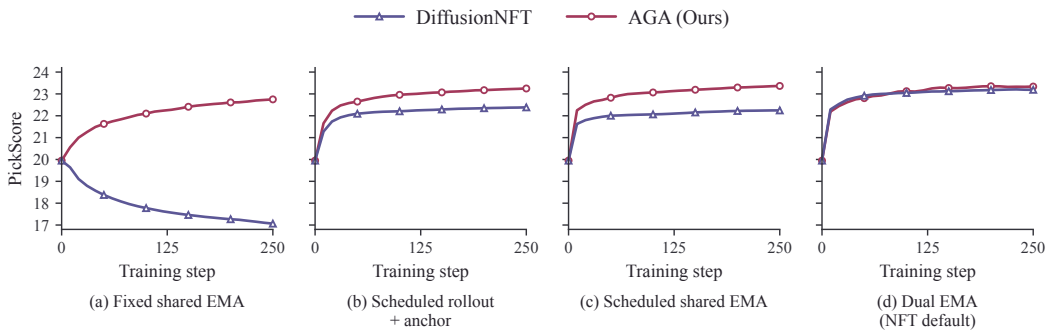

Figure 7: **Robustness across policy configurations.** Each panel compares AGA and DiffusionNFT under the same policy-copy update rules. Configuration (d) uses DiffusionNFT's official default.

imposing a stronger shared anchor, consistent with the cost of positive-sample retention analyzed in Section 3.3.

Efficient anchoring improves robustness to anchor-policy choices. Concentrating the correction on negative samples also reduces the number of updates that depend directly on an old prediction. This suggests a further benefit when the anchor policy changes. For a fixed sample, current prediction, and coefficient,

$$F_\omega(v; v'_{\text{anc}}) - F_\omega(v; v_{\text{anc}}) = \omega(v'_{\text{anc}} - v_{\text{anc}}). \tag{6}$$

AGA makes this dependence zero on positive samples and weights it by $|a|$ on negative samples. It thus retains the reference where the correction is useful while leaving positive fitting independent of the anchor prediction. We test the resulting robustness across four policy configurations drawn from previous implementations (Bergmeister et al., 2026; Zheng et al., 2026). These vary fixed or scheduled EMA updates and how rollout, anchor, and inference policies share their updates. Configuration (d) is DiffusionNFT's official PickScore default. The configurations and their sources are listed in Appendix D.

Figure 7 shows that AGA sustains PickScore gains across all four choices. DiffusionNFT is competitive under its default configuration (d), but improves more slowly and reaches lower rewards in (b) and (c), and loses reward in (a). AGA's effective use of the negative-sample anchor therefore extends beyond one policy-update setting. Its more consistent learning across these configurations supports the robustness benefit of preserving positive fitting while limiting direct anchor dependence to negative updates.

5 Conclusion

In advantage-weighted flow matching, increasing velocity error expresses negative feedback but does not directly measure suppression of an undesirable outcome. Our analysis identifies how policy anchoring can control the resulting update by counteracting negative curvature. Shared coefficient rules also change positive fitting, through either retention of old predictions or repulsion from them. This motivates separating the corrections applied to the two feedback branches. Experiments show that negative-sample anchoring uses this constraint more effectively, while additional repulsion on positive samples produces a pronounced reward–quality divergence followed by training collapse. Preserving ordinary positive fitting and correcting negative regression also leads to more consistent gains across policy configurations. AGA expresses these insights through the simple rule $-a\,\mathbf{1}[a < 0]$, using the old prediction only for negative feedback. Its performance across generation tasks demonstrates how a clearer understanding of anchoring can lead to a simpler and effective forward RL objective.

AI USE STATEMENT

We used GPT-5.6 Sol and GPT-6 Astra for the following purposes: (1) revising and polishing manuscript drafts; (2) assisting with LaTeX formatting, including generating Python scripts to convert experimental JSON outputs into LaTeX tables (Table1); and (3) conducting preliminary exploration of related work. All AI-assisted edits, scripts, generated tables, and literature findings were subsequently reviewed and verified by the authors before being incorporated into the manuscript.

REFERENCES

Joshua Achiam. Spinning up in deep reinforcement learning. OpenAI, 2018. URL https://spinningup.openai.com/en/latest/spinningup/rl_intro.html.

Andreas Bergmeister, Stefanie Jegelka, Nikolas Nüsken, Carles Domingo-Enrich, and Jakiw Pidstrigach. Reinforce adjoint matching: Scaling RL post-training of diffusion and flow-matching models. *arXiv preprint arXiv:2605.10759*, 2026.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2305.13301.

Huayu Chen, Kaiwen Zheng, Qinsheng Zhang, Ganqu Cui, Yin Cui, Haotian Ye, Tsung-Yi Lin, Ming-Yu Liu, Jun Zhu, and Haoxiang Wang. NFT: Bridging supervised learning and reinforcement learning in math reasoning. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/c9867d5a22653ce98b02595061e40f12-Abstract-Conference.html.

Jaemoo Choi, Wei Guo, Yuchen Zhu, Arash Vahdat, Molei Tao, Julius Berner, and Yongxin Chen. Scaling reinforcement learning for diffusion models via velocity matching. *arXiv preprint arXiv:2608.23664*, 2026a.

Jaemoo Choi, Yuchen Zhu, Wei Guo, Petr Molodyk, Bo Yuan, Jinbin Bai, Yi Xin, Molei Tao, and Yongxin Chen. Rethinking the design space of reinforcement learning for diffusion models: On the importance of likelihood estimation beyond loss design. In *International Conference on Machine Learning*, 2026b. URL https://arxiv.org/abs/2602.04663v2.

Kevin Clark, Paul Vicol, Kevin Swersky, and David Fleet. Directly fine-tuning diffusion models on differentiable rewards. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/13e8be77982beb73d7ed0bbf122f9f3c-Abstract-Conference.html.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12606–12633. PMLR, 2024. URL https://proceedings.mlr.press/v235/esser24a.html.

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 *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=2IoFFexvuw.

Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. DPOK: Reinforcement learning for fine-tuning text-to-image diffusion models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 79858–79885, 2023. doi: 10.52202/075280-3497. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/fc65fab891d83433bd3c8d966edde311-Abstract.html.

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.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. GenEval: An object-focused framework for evaluating text-to-image alignment. In *Advances in Neural Information Processing Systems*, volume 36, pp. 52132–52152, 2023. doi: 10.52202/075280-2270. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/a3bf71c7c63f0c3bcb7ff67c67b1e7b1-Abstract-Datasets_and_Benchmarks.html.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020. URL https://arxiv.org/abs/2006.11239.

Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-Pic: An open dataset of user preferences for text-to-image generation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 36652–36663, 2023. doi: 10.52202/075280-1594. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/73aacd8b3b05b4b503d58310b523553c-Abstract-Conference.html.

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.

Shufan Li, Konstantinos Kallidromitis, Akash Gokul, Yusuke Kato, and Kazuki Kozuka. Aligning diffusion models by optimizing human utility. In *Advances in Neural Information Processing Systems*, volume 37, pp. 24897–24925, 2024. doi: 10.52202/079017-0785. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/2c487f8a54cf24c0684c32abc77fed56-Abstract-Conference.html.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.02747.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-GRPO: Training flow matching models via online RL. In *Advances in Neural Information Processing Systems*, volume 38, pp. 40783–40818, 2025. doi: 10.52202/085713-1362. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/3a10c46572628d58cb44fb705f25cbbf-Abstract-Conference.html.

Zunxu Liu, Zhaofan Qiu, Yazhen Xie, Yingwei Pan, Ting Yao, and Tao Mei. On-policy diffusion reinforcement learning meets off-policy quality anchoring. In *European Conference on Computer Vision*, 2026. URL https://github.com/HiDream-ai/DiffusionCompass.

Julien Mairal. Incremental majorization-minimization optimization with application to large-scale machine learning. *SIAM Journal on Optimization*, 25(2):829–855, 2015. doi: 10.1137/140957639.

David McAllister, Songwei Ge, Brent Yi, Chung Min Kim, Ethan Weber, Hongsuk Choi, Haiwen Feng, and Angjoo Kanazawa. Flow matching policy gradients. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/3d43cc5692bf68944ee7cd31b97d0c11-Abstract-Conference.html.

Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1928–1937. PMLR, 2016. URL https://proceedings.mlr.press/v48/mniha16.html.

Zhenyang Ni, Yijiang Li, Ruochen Jiao, Simon Sinong Zhan, Sipeng Chen, Zhenfei Yin, Minshuo Chen, Philip Torr, Zhaoran Wang, and Qi Zhu. CreFlow: Corrective reflow for sparse-reward embodied video diffusion RL. *arXiv preprint arXiv:2605.14274*, 2026.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Raphael Gontijo-Lopes, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information*

Processing Systems, volume 35, pp. 36479–36494, 2022. doi: 10.52202/068431-2643. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/ec795aeadae0b7d230fa35cbaf04c041-Abstract-Conference.html.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Stability AI. Stable Diffusion 3.5 Medium. Model card, 2024. URL https://huggingface.co/stabilityai/stable-diffusion-3.5-medium.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8228–8238, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Wallace_Diffusion_Model_Alignment_Using_Direct_Preference_Optimization_CVPR_2024_paper.html.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. ImageReward: Learning and evaluating human preferences for text-to-image generation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 15903–15935, 2023. doi: 10.52202/075280-0700. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/33646ef0ed554145eab65f6250fab0c9-Abstract-Conference.html.

Yixian Xu, Yuanrui Zhang, Shengjie Luo, Liwei Wang, and Di He. Designing reinforcement learning for diffusion models: A unified path-space view. *arXiv preprint arXiv:2608.14430*, 2026.

Shuchen Xue, Chongjian Ge, Shilong Zhang, Yichen Li, and Zhi-Ming Ma. Advantage weighted matching: Aligning RL with pretraining in diffusion models. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2509.25050.

Alan L. Yuille and Anand Rangarajan. The concave-convex procedure. *Neural Computation*, 15(4): 915–936, 2003. doi: 10.1162/08997660360581958. URL https://www.cs.jhu.edu/~alanlab/Pubs03/yuille2003concave.pdf.

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*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/d8e68ddfe22520b45b8fb8d5cbde5a21-Abstract-Conference.html.

Yuanzhi Zhu, Xi Wang, Stéphane Lathuilière, and Vicky Kalogeiton. Diffusion reinforcement learning via centered reward distillation. *arXiv preprint arXiv:2603.14128*, 2026.