RECAST: REWARD CREDIT ASSIGNMENT ACROSS TIMESTEPS FOR ONLINE DIFFUSION REINFORCEMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Training diffusion models with multiple rewards requires distinguishing *user preference* from *reward informativeness*. User preference determines how much each reward should contribute to the overall objective; reward informativeness determines when its feedback is useful during denoising. Some rewards can meaningfully evaluate a sample as soon as global structure emerges, but others become informative only when the sample is nearly clean. To address both questions jointly, we propose ReCAST (**Re**ward **C**redit **AS**signment across **T**imesteps), the first method, to our knowledge, for per-reward, timestep-dependent credit assignment in diffusion reward fine-tuning. ReCAST separates user preferences from temporal allocation through a reward-by-timestep weight matrix $\boldsymbol{W}$, whose row sums match the user-specified reward budgets $\boldsymbol{\lambda}$, while its column sums are equal, assigning the same total weight to each denoising step. Under these marginal constraints, ReCAST allocates weight according to each reward's informativeness, quantified by its Rényi discriminability gain at each step. These gains telescope to the total discriminability between the reward-induced positive policy and the current policy, providing a basis for temporal credit assignment. We evaluate ReCAST by training SD3.5-Medium under two distinct four-reward settings, each across five reward budgets $\boldsymbol{\lambda}$. ReCAST improves the training rewards in one setting and matches them in the other, improves every held-out judge in both, and is preferred by an independent LLM-as-a-Judge. Overall, these results show that ReCAST achieves improvements that generalize beyond the training rewards and support its core principle: assigning each reward greater weight at the denoising timesteps where its feedback is most informative.

1 INTRODUCTION

Reward-based post-training has become a standard way to align diffusion and flow models with human intent, via policy gradient over the denoising chain (Black et al., 2023; Fan et al., 2023), backpropagation through a differentiable reward (Clark et al., 2023; Prabhudesai et al., 2024), preference optimization (Wallace et al., 2024; Zhu et al., 2025), group-relative objectives (Liu et al., 2025; Xue et al., 2025), or negative-aware fine-tuning on the forward process (Zheng et al., 2025). In practice, a single reward is rarely sufficient: standard training recipes mix prompt alignment (Hessel et al., 2021), learned human preference (Kirstain et al., 2023; Wu et al., 2023; Xu et al., 2023), and rule-based correctness (Ghosh et al., 2023) by linear scalarization $r(\boldsymbol{x}_0, \boldsymbol{c}) = \sum_i \lambda_i \, r_i(\boldsymbol{x}_0, \boldsymbol{c})$ with hand-set $\lambda_i \geq 0$, $\sum_i \lambda_i = 1$. Over-optimizing any single proxy can easily lead to reward hacking (Gao et al., 2023; Skalse et al., 2022).

However, static scalarization makes an implicit but imperfect assumption: once a reward is assigned an overall budget λ_i, its relative influence remains the same across all diffusion timesteps. Instead, we want a reward to have greater influence when it is more informative. Multi-reward alignment therefore involves two allocation problems: how much budget each reward receives, and when that budget should be spent. More specifically, given a user-specified reward budget $\boldsymbol{\lambda}$, can we allocate each reward's weights across timesteps more efficiently?

Our key intuition is that each reward measures a different visual property, and those properties do not become visible at the same point in the denoising trajectory. A reward for prompt alignment or rule-based correctness (Ghosh et al., 2023; Hessel et al., 2021) can already grade a sample once

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.

coarse layout and composition exist, while a reward for aesthetic or fine local detail cannot say much until the sample is nearly clean. Diffusion timesteps govern different aspects of generation by construction, with high-noise steps fixing global composition and low-noise steps refining local detail (Choi et al., 2022; Hang et al., 2023). A static, timestep-independent λ_i ignores this and forces every reward to contribute uniformly across t regardless of whether it can discriminate there. Although timestep-dependent treatment is well established for likelihood training (Karras et al., 2022; Kingma et al., 2021) and reward fine-tuning (He et al., 2025; Li et al., 2025; Liang et al., 2025), these methods apply a single reward-agnostic schedule, not the per-reward one we study.

Credit assignment is widely analyzed for language-model reasoning (Kazemnejad et al., 2025; Lightman et al., 2024; Uesato et al., 2022) but largely unexplored in the diffusion denoising chain. We propose ReCAST (**Re**ward **C**redit **AS**signment across **T**imesteps), the first per-reward timestep-dependent credit assignment method for diffusion reward fine-tuning, to our knowledge. More specifically, for rewards $r_1, \ldots, r_m$ and T steps, we replace the static convex weighting by an *automatically* designed weight matrix $\boldsymbol{W} \in \mathbb{R}_{\geq 0}^{m \times T}$, whose entry $W_{i,t}$ is the weight reward i carries at step t, and define the reward at timestep t as

$$r_t(\boldsymbol{x}_0, \boldsymbol{c}) \;=\; \sum_{i=1}^{m} T \cdot W_{i,t}\; r_i(\boldsymbol{x}_0, \boldsymbol{c}), \qquad \underbrace{\textstyle\sum_t W_{i,t} = \lambda_i}_{\text{inter-reward row budget}}\;, \qquad \underbrace{\textstyle\sum_i W_{i,t} = 1/T}_{\text{per-step column budget}}\;. \tag{1}$$

The row marginal $\lambda_i \geq 0$ ($\sum_i \lambda_i = 1$) is the inter-reward budget set by the user, and states *how much* reward i counts. The column marginal enforces that every step receives the same total weight $1/T$, so no step is starved or over-optimized. Together the two marginals fix the totals but not the shape: how row i spreads its budget λ_i across t is precisely *when* reward i counts, and that is what we estimate rather than hand-set. We first measure each reward's own per-step gain in Rényi discriminability along the trajectory (Sec. 3.2–3.3), then build a kernel from these gains and project it onto the feasible set of Eq. (1) to give the desired $\boldsymbol{W}^\star$ (Sec. 3.4).

2 Related work

RL in diffusion and flow models. Reward fine-tuning is central to current text-to-image models. DanceGRPO (Xue et al., 2025) extends group-relative optimization to both diffusion and rectified-flow image and video generators within a single framework; Flow-GRPO (Liu et al., 2025) converts the flow-matching ODE into an SDE so group sampling and its advantage estimator stay well defined; DiffusionNFT (Zheng et al., 2025) regresses implicit positive/negative velocity fields with a forward-only loss. Our work is built upon DiffusionNFT due to its performance and efficiency.

Combining multiple rewards. How multiple reward signals are combined has received wide attention in areas not limited to text-to-image tasks. SafeRLHF (Dai et al., 2023) decouples helpfulness and harmlessness into separate reward/cost models balanced via a Lagrange multiplier rather than a fixed scalar; Personalized Soups (Jang et al., 2023) instead trains one policy per preference dimension and merges them post-hoc, avoiding scalarization altogether, and Rewarded Soups (Ramé et al., 2023) takes the same merging route for reward proxies to reach Pareto trade-offs by interpolating weights. ALaRM (Lai et al., 2024) organizes rewards hierarchically rather than flattening them into one sum, and large reasoning models combine a rule-based outcome reward, a length penalty, and a language consistency reward within a single objective. GDPO (Liu et al., 2026) shows that summing rewards before GRPO's group normalization collapses into near-identical advantages, and fixes this by normalizing per reward before recombining. None of this work varies a reward's weight across denoising timesteps: our contribution is orthogonal to how the mixture itself is formed.

Timestep weighting in diffusion training. Existing works have adopted unequal treatment of denoising steps in diffusion training, such as ELBO-consistent weightings (Kingma et al., 2021; Song et al., 2021), perception-prioritized and Min-SNR schedules (Choi et al., 2022; Hang et al., 2023), EDM weighting (Karras et al., 2022), and importance sampling of t (Nichol & Dhariwal, 2021). On the reward side, step-by-step preference optimization (Liang et al., 2025) supervises every denoising step separately through a step-aware preference model, while MixGRPO's sliding ODE/SDE window (Li et al., 2025) and TempFlow-GRPO's noise-aware weighting (He et al., 2025) concentrate the reward signal on high-noise steps. All show that *when* a signal is applied matters,

but each shares one heuristic curve across rewards; ours is per-reward, from its own discriminability curve.

3 METHOD

This section builds the per-step gain in reward discriminability, then turns those gains into the weight matrix $\boldsymbol{W}$. In Sec. 3.2–3.3 we look at one reward, asking only how the per-step gain is shaped across t; Sec. 3.4 then puts the per-reward shapes into a matrix that satisfies both marginals of Eq. (1).

We work in the rectified-flow setup and with the DiffusionNFT objective, and put the background in App. A.1. Throughout this section we assume $r_i \geq 0$.

3.1 SETUP AND THE DENSITY RATIO

For each reward r_i, $i \in \{1, \dots, m\}$, let $\pi_i^+(\boldsymbol{x}_0 \mid \boldsymbol{c})$ be the positive policy induced by reward i on the old policy π^{old},

$$\pi_i^+(\boldsymbol{x}_0 \mid \boldsymbol{c}) := \frac{r_i(\boldsymbol{x}_0, \boldsymbol{c})}{\mathbb{E}_{\pi^{\text{old}}(\boldsymbol{x}_0|\boldsymbol{c})}[r_i(\boldsymbol{x}_0, \boldsymbol{c})]}\, \pi^{\text{old}}(\boldsymbol{x}_0 \mid \boldsymbol{c}) \tag{2}$$

and let $\pi_{i,t}^+(\boldsymbol{x}_t \mid \boldsymbol{c})$ and $\pi_t^{\text{old}}(\boldsymbol{x}_t \mid \boldsymbol{c})$ be the marginals at diffusion timestep t. Define the density ratio of r_i, $\rho_{i,t}(\boldsymbol{x}_t) := \pi_{i,t}^+(\boldsymbol{x}_t \mid \boldsymbol{c})/\pi_t^{\text{old}}(\boldsymbol{x}_t \mid \boldsymbol{c})$. Marginalizing the forward kernel and applying Bayes' rule, $\pi^{\text{old}}(\boldsymbol{x}_0 \mid \boldsymbol{x}_t, \boldsymbol{c}) = p(\boldsymbol{x}_t \mid \boldsymbol{x}_0)\pi^{\text{old}}(\boldsymbol{x}_0 \mid \boldsymbol{c}) \,/\, \pi_t^{\text{old}}(\boldsymbol{x}_t \mid \boldsymbol{c})$, gives the ratio at timestep t as a posterior expected reward normalized by the marginal one:

$$\rho_{i,t}(\boldsymbol{x}_t) \;:=\; \frac{\pi_{i,t}^+(\boldsymbol{x}_t \mid \boldsymbol{c})}{\pi_t^{\text{old}}(\boldsymbol{x}_t \mid \boldsymbol{c})} \;=\; \frac{\int p(\boldsymbol{x}_t \mid \boldsymbol{x}_0)\, \pi_i^+(\boldsymbol{x}_0 \mid \boldsymbol{c})\, \mathrm{d}\boldsymbol{x}_0}{\pi_t^{\text{old}}(\boldsymbol{x}_t \mid \boldsymbol{c})} \;=\; \frac{\mathbb{E}_{\pi^{\text{old}}(\boldsymbol{x}_0|\boldsymbol{x}_t,\boldsymbol{c})}\big[r_i(\boldsymbol{x}_0, \boldsymbol{c})\big]}{\mathbb{E}_{\pi^{\text{old}}(\boldsymbol{x}_0|\boldsymbol{c})}\big[r_i(\boldsymbol{x}_0, \boldsymbol{c})\big]}. \tag{3}$$

Everything on the right-hand side is a reward evaluation under the current policy, so the ratio is theoretically computable without ever directly sampling from π_i^+. For simplicity, we write $\mu_{i,t}(\boldsymbol{x}_t) := \mathbb{E}_{\pi^{\text{old}}(\boldsymbol{x}_0|\boldsymbol{x}_t,\boldsymbol{c})}[r_i]$ for the conditional reward and $Z_i := \mathbb{E}_{\pi^{\text{old}}(\boldsymbol{x}_0|\boldsymbol{c})}[r_i]$ for its marginal, so $\rho_{i,t}(\boldsymbol{x}_t) = \mu_{i,t}(\boldsymbol{x}_t)/Z_i$.

3.2 REWARD DISCRIMINABILITY

We first ask one question: how distinguishable is the positive policy from the current one at a given timestep? We measure this by the Rényi divergence (Rényi, 1961) of order $\alpha > 1$. The *cumulative reward discriminability* at timestep t for reward i is therefore

$$D_{i,t} \;:=\; D_\alpha\big(\pi_{i,t}^+ \,\big\|\, \pi_t^{\text{old}}\big) \;=\; \frac{1}{\alpha - 1} \log \mathbb{E}_{\pi_{i,t}^+}\Big[\rho_{i,t}(\boldsymbol{x}_t)^{\alpha-1}\Big]. \tag{4}$$

It vanishes exactly when the two marginals agree, grows as reward i separates them, and is nondecreasing in α. Its $\alpha \to 1^+$ limit is the KL divergence between the same two marginals,

$$\lim_{\alpha \to 1^+} D_{i,t} \;=\; \mathrm{KL}\big(\pi_{i,t}^+ \,\big\|\, \pi_t^{\text{old}}\big) \;=\; \mathbb{E}_{\pi_{i,t}^+}\big[\log \rho_{i,t}(\boldsymbol{x}_t)\big]. \tag{5}$$

Sec. 3.3 shows why we choose Rényi instead of the KL divergence.

At pure noise ($t = T$) the forward map has $\sigma_T = 1$ and discards $\boldsymbol{x}_0$ entirely, so $\pi_{i,T}^+ = \pi_T^{\text{old}} = \mathcal{N}(0, I)$ and $D_{i,T} = 0$ for every reward. At clean data the divergence reaches its maximum $D_{i,0} = D_\alpha(\pi_{i,0}^+ \,\|\, \pi_0^{\text{old}})$, the full data-level discriminability reward i induces.

Discriminability gain. Cumulative discriminability $D_{i,t}$ contains all signal accumulated from T to t, so weighting by it would repeatedly count earlier gains. Therefore, we use the per-step *gain*

$$\Delta D_{i,t} := D_{i,t-1} - D_{i,t}, \tag{6}$$

which isolates the new discriminability contributed by reward i at transition $t \to t-1$. The gains telescope along the denoising trajectory:

$$\sum_{t=1}^{T} \Delta D_{i,t} = D_{i,0} - D_{i,T} = D_{i,0} = D_\alpha\big(\pi_{i,0}^+ \,\big\|\, \pi_0^{\text{old}}\big). \tag{7}$$

Thus, $\Delta D_{i,t}$ decomposes the reward's total data-level discriminability across timesteps without double-counting: rewards sensitive to global structure concentrate at large t, while those sensitive to fine details concentrate at small t.

One basic property is that $\Delta D_{i,t} \geq 0$, so at each denoising step ($t \to t-1$) the discriminability does not decrease. This ensures the soundness of our definition of $\Delta D_{i,t}$. We prove a general theorem.

Theorem 3.1 (Rényi dissipation under shared diffusion)**.** *Let π_t^a and π_t^b be positive, sufficiently smooth probability densities on $\mathbb{R}^d$ that both evolve under the same forward Fokker–Planck equation*

$$\partial_t \pi_t \;=\; -\nabla\cdot\big(\boldsymbol{u}(\boldsymbol{x},t)\,\pi_t\big) \;+\; \tfrac{g(t)^2}{2}\,\Delta\pi_t, \tag{8}$$

for a shared drift $\boldsymbol{u}$ and spatially constant diffusion coefficient $g(t)$. Assume the displayed moments and derivatives are integrable and that the boundary terms in the integrations by parts vanish. Let $\rho := \pi^a/\pi^b$ denote the density ratio, and define the Rényi-tilted distribution *of order α:*

$$\pi_t^{(\alpha)}(\boldsymbol{x}) \;:=\; \frac{\rho(\boldsymbol{x})^\alpha}{\mathbb{E}_{\pi_t^b}[\rho^\alpha]}\,\pi_t^b(\boldsymbol{x}) \;\propto\; \big(\pi_t^a\big)^\alpha\,\big(\pi_t^b\big)^{1-\alpha}. \tag{9}$$

This geometric family recovers π_t^b at $\alpha = 0$ and π_t^a at $\alpha = 1$; in the regime used here, $\alpha > 1$, it extrapolates beyond π_t^a. Then for every $\alpha > 0$ with $\alpha \neq 1$,

$$\frac{\mathrm{d}}{\mathrm{d}t}\,D_\alpha\big(\pi_t^a \,\|\, \pi_t^b\big) \;=\; -\,\frac{g(t)^2}{2}\cdot\alpha\cdot\mathbb{E}_{\pi_t^{(\alpha)}}\big[\|\nabla\log\tfrac{\pi^a}{\pi^b}\|^2\big], \tag{10}$$

where $\mathbb{E}_{\pi_t^{(\alpha)}}[\|\nabla\log\frac{\pi^a}{\pi^b}\|^2]$ is the Fisher information of the log-density ratio measured under the tilted distribution $\pi_t^{(\alpha)}$.

The proof of the theorem is deferred to App. A.2. Both $\pi_{i,t}^+$ and π_t^{old} evolve according to Eq. (8) with the same coefficients because the diffusion kernel $p(\boldsymbol{x}_t \mid \boldsymbol{x}_0)$ is independent of the policy. Away from the singular endpoint $\sigma_t = 1$, the forward process $\boldsymbol{x}_t = (1-\sigma_t)\boldsymbol{x}_0 + \sigma_t\boldsymbol{\epsilon}$ is the transition law of a linear SDE, and therefore satisfies Eq. (8) with $\boldsymbol{u}(\boldsymbol{x},t) = -\frac{\dot\sigma_t}{1-\sigma_t}\boldsymbol{x}$, $\quad g(t)^2 = \frac{2\sigma_t\dot\sigma_t}{1-\sigma_t}$. Since Eq. (8) is linear in π_t, and the reward tilt r_i acts only on the initial state $\boldsymbol{x}_0$, the tilt changes only the initial distribution rather than the evolution coefficients. Hence, $\pi_{i,t}^+$ and π_t^{old} follow the same Fokker–Planck dynamics from different initial conditions. Theorem 3.1 therefore applies with $\pi^a = \pi_{i,t}^+$, $\pi^b = \pi_t^{\mathrm{old}}$, and $\rho = \rho_{i,t}$, yielding $\frac{\mathrm{d}}{\mathrm{d}t}D_\alpha\big(\pi_{i,t}^+ \,\|\, \pi_t^{\mathrm{old}}\big) \leq 0$.

3.3 Estimation details

Why we use $\alpha > 1$ rather than the KL limit. The exact KL in Eq. (5) is well defined under the usual absolute-continuity and integrability conditions: states with $\rho_{i,t} = \pi_{i,t}^+/\pi_t^{\mathrm{old}} = 0$ have zero probability under $\pi_{i,t}^+$ and therefore do not cause the expectation to diverge. However, empirically in finite-sample rollouts, the estimated conditional reward $\hat{\mathbb{E}}_{\pi^{\mathrm{old}}(\boldsymbol{x}_0\mid\boldsymbol{x}_t,\boldsymbol{c})}\big[r_i(\boldsymbol{x}_0,\boldsymbol{c})\big]$, and hence $\hat\rho_{i,t}$, can be exactly zero, in which case averaging $\log\hat\rho_{i,t}$ introduces $-\infty$ values. For example, consider a noisy state $\boldsymbol{x}_t$ that has already committed to a two-object layout when the prompt requires only one. Such an $\boldsymbol{x}_t$ will likely be denoised under π^{old} into samples $\boldsymbol{x}_0$ that keep that structure, so $r_i(\boldsymbol{x}_0,\boldsymbol{c}) = 0$ for multiple $\boldsymbol{x}_0$. Simply swapping the two marginals does not resolve this issue: $\mathrm{KL}\big(\pi_t^{\mathrm{old}} \,\big\|\, \pi_{i,t}^+\big) = -\mathbb{E}_{\pi_t^{\mathrm{old}}}\big[\log\rho_{i,t}(\boldsymbol{x}_t)\big]$ may also diverge for the same reason.

This motivates using the Rényi divergence with $\alpha > 1$. In this case, the logarithm in Eq. (4) is applied only after the empirical moment $\hat{\mathbb{E}}_{\pi_{i,t}^+}\big[\hat\rho_{i,t}(\boldsymbol{x}_t)^{\alpha-1}\big]$. Consequently, a zero conditional reward estimate, corresponding to a state $\boldsymbol{x}_t$ for which $\hat\rho_{i,t}(\boldsymbol{x}_t) = 0$, contributes zero to the empirical moment rather than producing an infinite logarithmic term. The estimate therefore remains finite whenever the empirical moment itself is positive, which is usually satisfied in practical estimation. At the same time, D_α retains a useful relation to the original objective: because Rényi divergence is nondecreasing in its order, D_α upper-bounds the KL for $\alpha > 1$. App. B.3 examines how the resulting discriminability curve varies with α. We use $\alpha = 2$ in all training experiments for its simplicity: the exponent in Eq. (4) is then $\alpha - 1 = 1$.

Sampling from π^+ for a higher-contrast surrogate. Revisiting Eq. (4), we find one problem: sampling $\boldsymbol{x}_t$ from $\pi_{i,t}^+$ is impractical since we cannot directly sample from the π_i^+ of Eq. (2). One

Algorithm 1 Per-step Rényi discriminability gain proxies $\widehat{\widetilde{\Delta D}}_{i,t}$ from π^+ samples

Require: Prompt $\boldsymbol{c}$, policy π^{old}, stronger external generator $\pi^+(\cdot \mid \boldsymbol{c})$ used as a common proposal, rewards $r_{1:m}$, order $\alpha > 0$ with $\alpha \neq 1$, samples N, rollouts K, schedule $\{\sigma_t\}_{t=0}^{T}$

Ensure: Per-step gain proxies $\widehat{\widetilde{\Delta D}}_{i,t}$ for every reward i and step t

for $n = 1, \ldots, N$ **do**
  Draw $\widetilde{\boldsymbol{x}}_0^{(n)} \sim \pi^+(\cdot \mid \boldsymbol{c})$; set $\boldsymbol{x}_t^{(n)} \leftarrow (1-\sigma_t)\widetilde{\boldsymbol{x}}_0^{(n)} + \sigma_t \boldsymbol{\epsilon}_t$, $\boldsymbol{\epsilon}_t \sim \mathcal{N}(0, I)$, $t = 0, \ldots, T$ // $\boldsymbol{x}_t^{(n)} \sim \pi_t^+$
  $\hat{\mu}_{i,t}^{(n)} \leftarrow \frac{1}{K_t} \sum_{k \le K_t} r_i(\boldsymbol{x}_0^{(n,t,k)}, \boldsymbol{c})$ from K_t rollouts $\boldsymbol{x}_t^{(n)} \to \boldsymbol{x}_0^{(n,t,k)}$ under π^{old} (K_0=1, else K)
end for
$\widetilde{D}_{i,t} \leftarrow \frac{1}{\alpha-1} \log\big(\max\{10^{-10}, \frac{1}{N}\sum_n [\hat{\mu}_{i,t}^{(n)}]^{\alpha-1}\}\big)$ for all i, t // *numerical floor*
return $\widehat{\widetilde{\Delta D}}_{i,t} \leftarrow \big[\widehat{\widetilde{D}}_{i,t-1} - \widehat{\widetilde{D}}_{i,t}\big]_+$, $t = 1, \ldots, T$ // *Eq.* (12)*, clipped*

exact alternative is to sample from π^{old} instead: since $\rho_{i,t} = \pi_{i,t}^+ / \pi_t^{\text{old}}$, we have $\mathbb{E}_{\pi_t^{\text{old}}}[\rho_{i,t}(\boldsymbol{x}_t)^\alpha] = \mathbb{E}_{\pi_{i,t}^+}\big[\rho_{i,t}(\boldsymbol{x}_t)^{\alpha-1}\big]$. This means we draw $\widetilde{\boldsymbol{x}}_0 \sim \pi^{\text{old}}(\cdot \mid \boldsymbol{c})$, forward-noise it to $\boldsymbol{x}_t \sim \pi_t^{\text{old}}$ by $\boldsymbol{x}_t = (1-\sigma_t)\widetilde{\boldsymbol{x}}_0 + \sigma_t \boldsymbol{\epsilon}$, and roll out multiple $\boldsymbol{x}_0$ from $\boldsymbol{x}_t$ under π^{old}. Here, $\boldsymbol{x}_0$ is used to estimate $\mu_{i,t}(\boldsymbol{x}_t)$ and $\widetilde{\boldsymbol{x}}_0$ is used to estimate Z_i. However, when a reward has small relative variance under π^{old}, the exact divergence is correspondingly small: at $\alpha = 2$, $D_{i,0} = \log(1 + \mathrm{CV}_{\pi^{\text{old}}}(r_i)^2)$. Near-zero curves therefore indicate low reward variability, and finite-sample differencing can yield noisy clipped profiles (Fig. 3; App. B.4).

Therefore, to obtain a higher-contrast profile at the available budget, we use a stronger external generator π^+ as a common surrogate: we draw $\widetilde{\boldsymbol{x}}_0 \sim \pi^+(\cdot \mid \boldsymbol{c})$, forward-noise it to $\boldsymbol{x}_t \sim \pi_t^+$, and roll out $\boldsymbol{x}_0$ from $\boldsymbol{x}_t$ under π^{old}. This produces smoother profiles at the same budget (App. B.4). Because the ratio $\pi_{i,t}^+ / \pi_t^+$ is unknown, this is not an exact importance-sampling estimator of $D_{i,t}$ but a surrogate log-moment. App. A.3 analyzes the error the surrogate induces: when $\mathrm{KL}(\pi^+ \| \pi_i^+)$ is small, the error in the discriminability gain is $O(\sqrt{\mathrm{KL}(\pi^+ \| \pi_i^+)})$ and propagates to the Sinkhorn weights. Furthermore, replacing π^+ with π^{old} self-rollouts yields far noisier curves in App. B.4, while choosing a different π^+, either GPT IMAGE 1.5 or NANO BANANA PRO, leaves the estimated curves essentially unchanged in App. B.5.

Denominator cancellation. Putting $\rho_{i,t}(\boldsymbol{x}_t) = \mu_{i,t}(\boldsymbol{x}_t)/Z_i$ into Eq. (4), we have $D_{i,t} = \frac{1}{\alpha-1} \log \mathbb{E}_{\pi_{i,t}^+}[\mu_{i,t}(\boldsymbol{x}_t)^{\alpha-1}] - \log Z_i$. Because only the *gain* $\Delta D_{i,t} = D_{i,t-1} - D_{i,t}$ is needed, $\log Z_i$ cancels entirely:

$$\boxed{\Delta D_{i,t} = \frac{1}{\alpha-1}\Big(\log \mathbb{E}_{\pi_{i,t-1}^+}\big[\mu_{i,t-1}(\boldsymbol{x}_{t-1})^{\alpha-1}\big] - \log \mathbb{E}_{\pi_{i,t}^+}\big[\mu_{i,t}(\boldsymbol{x}_t)^{\alpha-1}\big]\Big).} \tag{11}$$

Replacing the unknown positive marginals by the common proposal π_t^+ leaves the quantity we actually compute,

$$\widetilde{\Delta D}_{i,t} = \frac{1}{\alpha-1}\Big(\log \mathbb{E}_{\pi_{t-1}^+}\big[\mu_{i,t-1}(\boldsymbol{x}_{t-1})^{\alpha-1}\big] - \log \mathbb{E}_{\pi_t^+}\big[\mu_{i,t}(\boldsymbol{x}_t)^{\alpha-1}\big]\Big), \tag{12}$$

which coincides with Eq. (11) only when $\pi_t^+ = \pi_{i,t}^+$. Only $\mu_{i,t}$ has to be estimated, by rollouts under π^{old}, and Z_i is never formed. Alg. 1 is the resulting estimator, in compact form; App. B.1 writes out every index. For readability in the empirical sections we write the output $\widehat{\widetilde{\Delta D}}_{i,t}$ simply as $\Delta D_{i,t}$, and identities involving D_α refer to the population quantity unless stated otherwise.

3.4 SINKHORN PROJECTION

The gains $\Delta D_{i,t}$ measure the utility of each reward at every step, but they do not yet constitute valid weights: they neither satisfy the budget $\boldsymbol{\lambda}$ nor maintain equal per-step total weights across t. We formulate the two requirements as marginal constraints and single out a unique matrix satisfying them by entropic projection. We ablate the choice of equal per-step total weights in Sec. 4.5.

We first define the *mean-normalized gain*

$$\overline{\Delta D}_{i,t} \;:=\; \frac{\Delta D_{i,t}}{\frac{1}{T}\sum_{s=1}^{T}\Delta D_{i,s}}, \qquad \text{so that} \qquad \frac{1}{T}\sum_{t}\overline{\Delta D}_{i,t} \;=\; 1 \;\text{ for every } i. \tag{13}$$

Because the gains telescope (Eq. (7)), the denominator is $D_{i,0}/T$, so the division removes the reward's total discriminability and keeps only its shape in t to better capture the intra-timestep relation.

From the normalized gains we build the *affinity kernel*

$$K_{i,t} \;:=\; \exp\big(\overline{\Delta D}_{i,t}\big) \tag{14}$$

which is largest at the steps where reward i contributes most of its discriminability and carries no scale parameter of its own, since $\overline{\Delta D}_{i,t}$ is already mean-normalized. Theorem 3.1 gives $\Delta D_{i,t} \geq 0$ in population; in finite samples we clip, using $[\Delta D_{i,t}]_+$, so a step estimated as negative enters at $K_{i,t} = 1$, the smallest value the kernel takes.

Recall the two marginal constraints from Eq. (1). Writing $\mathbf{1}_m, \mathbf{1}_T$ for all-ones vectors, the feasible set is the *transportation polytope*

$$\mathcal{U}(\boldsymbol{\lambda}) \;:=\; \Big\{\, \boldsymbol{W} \in \mathbb{R}^{m\times T}_{\geq 0} \;:\; \boldsymbol{W}\mathbf{1}_T = \boldsymbol{\lambda},\;\; \boldsymbol{W}^\top \mathbf{1}_m = \tfrac{1}{T}\mathbf{1}_T \,\Big\}. \tag{15}$$

The set is never empty, since the rank-one matrix λ_i/T always belongs to it. Among its elements we want the KL projection of $\boldsymbol{K}$ onto the polytope,

$$\boldsymbol{W}^\star \;=\; \operatorname*{arg\,min}_{\boldsymbol{W}\in\mathcal{U}(\boldsymbol{\lambda})} \mathrm{KL}\big(\boldsymbol{W} \,\|\, \boldsymbol{K}\big) \;=\; \operatorname*{arg\,min}_{\boldsymbol{W}\in\mathcal{U}(\boldsymbol{\lambda})} \sum_{i,t} W_{i,t}\log\frac{W_{i,t}}{K_{i,t}} - W_{i,t} + K_{i,t}, \tag{16}$$

which, substituting Eq. (14), is an entropy-regularized transport problem with cost $C_{i,t} := -\overline{\Delta D}_{i,t}$ and unit regularization strength:

$$\boldsymbol{W}^\star \;=\; \operatorname*{arg\,min}_{\boldsymbol{W}\in\mathcal{U}(\boldsymbol{\lambda})} \sum_{i,t} \Big[\, C_{i,t}\, W_{i,t} \;+\; W_{i,t}\log W_{i,t} \,\Big]. \tag{17}$$

Since every $K_{i,t} > 0$, the objective is strictly convex and the minimizer is unique. Setting the Lagrangian gradient to zero gives the kernel rescaled by one factor per row and one per column,

$$W^\star_{i,t} \;=\; a_i\, K_{i,t}\, b_t \;=\; a_i \cdot \exp\big(\overline{\Delta D}_{i,t}\big) \cdot b_t, \tag{18}$$

where a_i carries inter-reward scaling, and b_t the per-step normalization. Sinkhorn's theorem (Cuturi, 2013; Sinkhorn & Knopp, 1967) guarantees that $\boldsymbol{a} \in \mathbb{R}^m_{>0}$ and $\boldsymbol{b} \in \mathbb{R}^T_{>0}$ exist and are unique up to the trivial rescaling $(\boldsymbol{a}, \boldsymbol{b}) \mapsto (c\,\boldsymbol{a}, \boldsymbol{b}/c)$, and it finds them iteratively:

$$a_i \;\leftarrow\; \frac{\lambda_i}{\sum_t K_{i,t}\, b_t}, \qquad\qquad b_t \;\leftarrow\; \frac{1/T}{\sum_i K_{i,t}\, a_i}, \tag{19}$$

which converges linearly; in the log domain it needs < 100 steps at $m \leq 4, T \leq 25$ (Alg. 3, App. C).

A reward whose gain is flat in t has $\overline{\Delta D}_{i,t} \equiv 1$, since Eq. (13) fixes the mean at 1; if this holds for every reward, the kernel is rank one and the entropic optimum is $W^\star_{i,t} = \lambda_i/T$, so the static baseline is exactly the case of flat estimated curves (App. C.2).

3.5 Empirical gain curves on SD3.5-Medium

Setup. We instantiate Alg. 1 on SD3.5-Medium with α=2, 80 prompts from the training datasets, N=8 samples per prompt, and K=16 rollouts per $\boldsymbol{x}_t$. We approximate π^+ by GPT Image 1.5, VAE-encode its images into latent space, forward-noise to each step, and run batched rollouts under π^{old}. We plot seven rewards: CLIPScore, HPSv2, PickScore, Aesthetic, and ImageReward jointly on this shared T=10 grid, plus OCR and GenEval on their own prompt sets at the native T=25 schedule they are later trained under. Each reward's architecture, checkpoint, and range are in Sec. 4.1 and App. E.1; App. B.3 conducts an ablation study on α.

Results. In Fig. 1, the estimated curves are strongly reward-specific; the horizontal axis is the diffusion timestep, from the cleanest step t=1 (left) to the noisiest t=T (right). GenEval, OCR, and ImageReward peak the most sharply, concentrating their gain toward the noisy end, while the per-step gain of the others stays close to uniform. The first two are rule-based and, like ImageReward, can score a sample once its global structure has emerged. The differences are large

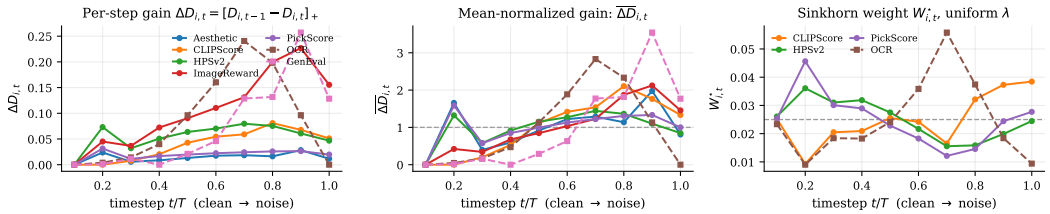


Figure 1: Gain curves and the weight matrix they project to. **Left:** per-step gain $\Delta D_{i,t}$. **Center:** mean-normalized gain $\overline{\Delta D}_{i,t}$ (Eq. (13)), the exponent of the affinity kernel (Eq. (14)). **Right:** the Sinkhorn matrix $W^{\star}_{i,t}$ (Eq. (18)) at uniform $\lambda_i{=}1/m$, solved over the four rewards that are trained together in the OCR setting; both marginals hold exactly, every column summing to $1/T$ and every row to λ_i. In all panels the horizontal axis is t/T, running from the cleanest step $t{=}1$ (left) to the noisiest $t{=}T$ (right). OCR and GENEVAL (dashed) are resampled onto the shared $T{=}10$ grid from their native $T{=}25$ curves, so the right panel is illustrative: the training-stage kernel is solved at $T{=}25$ on OCR's own grid (Sec. 4.1).

enough to matter: under uniform $\boldsymbol{\lambda}$, the aggregate demand $\sum_i \lambda_i \overline{\Delta D}_{i,t}$ varies by $5.4\times$ across the steps $2 \le t \le T$, and still by $4.5\times$ once GENEVAL is excluded. The right panel shows the Sinkhorn projection (Eq. (19)) over the four rewards in the OCR setting, correcting this imbalance by shifting each reward's weight toward its own high-gain steps while both marginals stay exact.

4 EXPERIMENTS

Sec. 4.1 introduces the setup. Sec. 4.2 evaluates the training rewards on a held-out split, and Sec. 4.3 the same checkpoints under held-out judges, to show generalizable improvement. Sec. 4.4 rescores those images with an independent LLM-as-a-Judge, and Sec. 4.5 ablates the Sinkhorn projection.

4.1 TRAINING SETUP

Reward models. We evaluate ReCAST with nine reward models in two distinct roles. Five are *training rewards*: CLIPSCORE (Hessel et al., 2021), HPSV2 (Wu et al., 2023), PICKSCORE (Kirstain et al., 2023), OCR, and GENEVAL (Ghosh et al., 2023). To evaluate generalization, four *held-out judges* score the checkpoints: AESTHETIC (Schuhmann, 2022), IMAGEREWARD (Xu et al., 2023), HPSV3 (Ma et al., 2025), and UNIFIEDREWARD-2.0 (Wang et al., 2025). App. E.1 details architectures, checkpoints, and output ranges for all rewards.

Training data and reward chain. Following DiffusionNFT (Zheng et al., 2025), training runs in two stages. The **warmup stage** jointly optimizes CLIPSCORE, HPSV2, and PICKSCORE for 120 steps from the base model on Pick-a-Pic prompts (25,432 train/2,048 test); every later run resumes from its checkpoint. The **training stage-OCR** then adds an OCR reward on text-rendering prompts (19,652 train/1,017 test) for 60 further steps ($m{=}4$). The **training stage-GenEval** branches from the same warmup parent on compositional GENEVAL prompts (50,000 train/2,211 test), swapping OCR for a rule-based GENEVAL reward on an identical schedule.

Training configuration. Across all stages, we fine-tune LoRA adapters of Stable Diffusion 3.5-Medium. We use the DiffusionNFT objective with $\beta{=}0.1$, rolling out $T{=}25$ sampling steps per iteration at an effective batch size of 1152. We run the warmup stage at a balanced $\boldsymbol{\lambda} = (1,1,1)$ on three rewards: CLIPSCORE, HPSV2, and PICKSCORE. Then, we sweep each training-stage setting's row marginals over $\boldsymbol{\lambda} \in \{(1,1,1,1),(1,1,1,2),(1,1,2,1),(1,2,1,1),(2,1,1,1)\}$, the uniform budget together with each coordinate doubled in turn, where $\boldsymbol{\lambda}$ is ordered (CLIPSCORE, HPSV2, PICKSCORE, OCR/GENEVAL). Every $\boldsymbol{\lambda}$ is written unnormalized as an integer ratio between rewards, and is divided by its own sum before use to satisfy $\sum_i \lambda_i = 1$. Each budget gives one matched pair of ReCAST against a static baseline, and both settings are repeated end-to-end under 3 independent seeds, giving 15 matched (budget, seed) pairs per setting. For our weights configuration, we use Alg. 1 with $\alpha{=}2$ for the kernel and Alg. 3 for the Sinkhorn projection. The training objective is detailed in App. A.1 and the remaining hyperparameters in App. E.2.

Evaluation protocol. We evaluate at three levels: the **training rewards** on a held-out dataset (Sec. 4.2); **held-out judge models** (Sec. 4.3); and **LLM-as-a-Judge** (Sec. 4.4). All three score each run's final checkpoint at matched seeds, 512^2 resolution, CFG 4.5, and 40 steps; we report means over each training-stage pair-set and win rates over its matched pairs.

4.2 TRAINING REWARDS

Table 1: Training-reward scores at the final checkpoint, reported as base / static / Ours, where *base* is the untuned SD3.5-Medium, *static* is static reweighting by $\boldsymbol{\lambda}$, and Aggregate r is the sum of the individual rewards weighted by the integer $\boldsymbol{\lambda}$. Both settings were trained end-to-end (warmup stage + training-stage sweep, both methods) under 3 seeds; entries are the mean of the three seed-level means, standard deviation as a subscript, and *win* pools all 15 matched (budget, seed) pairs. Bold marks the better of static and Ours, and both when the two agree at the reported precision. Per-$\boldsymbol{\lambda}$ results are in App. D.

	Training stage-OCR (5 budgets × 3 seeds)		**Training stage-GenEval** (5 budgets × 3 seeds)	
Training reward	base / static / Ours	win	base / static / Ours	win
Aggregate r	1.642 / $2.867_{\pm 0.026}$ / **2.920**$_{\pm 0.028}$	11/15	1.784 / **2.881**$_{\pm 0.021}$ / $2.878_{\pm 0.015}$	7/15
CLIPSCORE	0.271 / $0.313_{\pm 0.002}$ / **0.320**$_{\pm 0.002}$	13/15	0.240 / **0.299**$_{\pm 0.001}$ / **0.299**$_{\pm 0.001}$	6/15
HPSV2	0.195 / $0.288_{\pm 0.008}$ / **0.303**$_{\pm 0.004}$	12/15	0.213 / **0.313**$_{\pm 0.005}$ / $0.312_{\pm 0.002}$	5/15
PICKSCORE	0.773 / $0.888_{\pm 0.002}$ / **0.899**$_{\pm 0.003}$	15/15	0.790 / **0.911**$_{\pm 0.004}$ / $0.908_{\pm 0.003}$	6/15
Target reward	0.129 / $0.893_{\pm 0.014}$ / **0.900**$_{\pm 0.032}$	9/15	0.244 / $0.873_{\pm 0.011}$ / **0.874**$_{\pm 0.007}$	9/15

Tab. 1 reports each training reward on a held-out dataset, averaged over the five budgets and three seed replicates, with *win* counting the matched pairs in which ReCAST beats static. ReCAST raises the aggregate score from 2.867 to 2.920 on average ($+0.053_{\pm 0.049}$ across seeds) and wins 11 of the 15 matched (budget, seed) pairs. The gain is not obtained by sacrificing one objective for another: all four component rewards improve on average, with PICKSCORE winning all 15 pairs and CLIPSCORE and HPSV2 winning 13 and 12. The target OCR reward carries the largest seed variance ($+0.007_{\pm 0.046}$, $9/15$). Adaptive timing therefore uses a fixed multi-reward budget more efficiently, rather than merely changing the trade-off encoded by $\boldsymbol{\lambda}$. On the GenEval setting the two methods instead tie: every training reward shifts by less than one seed standard deviation (aggregate $-0.003_{\pm 0.031}$, target GENEVAL $+0.001_{\pm 0.017}$) and win rates are near chance (5 to 9 of 15). That setting therefore acts as a matched-training-reward control, where Sec. 4.3 still finds ReCAST preferred by every held-out judge.

4.3 HELD-OUT JUDGES

Table 2: Held-out judge scores at the final checkpoint, reported as base / static / Ours. Both settings cover 5 budgets × 3 independent training seeds, and entries, subscripts, *win*, and the bolding rule are as in Tab. 1; the HPSV3 row is at two decimals because its scale is an order of magnitude larger. Per-$\boldsymbol{\lambda}$ results are in App. D.

	Training stage-OCR (15 pairs)		**Training stage-GenEval** (15 pairs)	
Held-out model	base / static / Ours	win	base / static / Ours	win
IMAGEREWARD	1.140 / $1.422_{\pm 0.037}$ / **1.482**$_{\pm 0.030}$	13/15	1.024 / $1.198_{\pm 0.165}$ / **1.289**$_{\pm 0.122}$	9/15
AESTHETIC	5.388 / $5.486_{\pm 0.063}$ / **5.578**$_{\pm 0.009}$	12/15	5.949 / $5.864_{\pm 0.120}$ / **5.968**$_{\pm 0.102}$	10/15
HPSV3	12.69 / $12.74_{\pm 0.18}$ / **13.14**$_{\pm 0.49}$	11/15	9.19 / $8.02_{\pm 0.95}$ / **8.58**$_{\pm 0.56}$	10/15
UNIFIEDREWARD-2.0	3.015 / $3.154_{\pm 0.025}$ / **3.176**$_{\pm 0.040}$	11/15	2.682 / $2.671_{\pm 0.085}$ / **2.705**$_{\pm 0.046}$	9/15

As shown in Tab. 2, on the OCR setting all four held-out judges score ReCAST higher on average, winning 13, 12, 11, and 11 of the 15 matched (budget, seed) pairs. For the GenEval setting, although the two methods are tied there on every reward being optimized (Sec. 4.2), all four held-out judges still favor ReCAST (largest, HPSV3 $+0.56_{\pm 0.41}$), winning 9 or 10 of the 15 pairs. A held-out gain under an equal training reward is consistent with generalization rather than overfitting, and on the OCR setting the improved text rendering enhances rather than degrades broader image quality. The two settings differ against the untuned base model, however: the OCR setting clears it on all four judges under both methods, whereas in the GenEval setting static falls below it on AESTHETIC, HPSV3, and UNIFIEDREWARD-2.0. Optimizing compositional correctness therefore costs generic visual quality, and ReCAST recovers that cost on every judge, fully on AESTHETIC and UNIFIEDREWARD-2.0 and partly on HPSV3.

4.4 LLM-AS-A-JUDGE

To assess general preference closer to real use, we adapt the MMRB2 image-generation judge prompt and forward–reverse evaluation protocol (Hu et al., 2025) to compare the ReCAST checkpoints against static on a fixed 1,000-prompt subset of the held-out OCR set. We serve the resulting multimodal LLM judge with `gemini-3.5-flash` at temperature 0. It picks the better of the two images per prompt according to the fixed rubric, and every pair is judged twice with positions swapped to cancel position bias; we also report the rubric's faithfulness and aesthetics criteria. App. E.3 gives the full prompt and the debiasing procedure.

Table 3: Pairwise evaluation of ReCAST against static using the MMRB2 judge protocol on the same subset as Tab. 2, position-debiased over both presentation orders. Every entry is ReCAST's win rate, overall and under the two rubric criteria, with 0.5 indicating parity; $\boldsymbol{\lambda}$ is ordered (CLIPSCORE, HPSV2, PICKSCORE, OCR).

$\boldsymbol{\lambda}$	$(1,1,1,1)$	$(1,1,1,2)$	$(1,1,2,1)$	$(1,2,1,1)$	$(2,1,1,1)$	**mean**
Overall	0.6538	0.5580	0.4780	0.5365	0.6325	**0.5718**
Faithfulness	0.6550	0.5570	0.5130	0.5655	0.6094	**0.5800**
Aesthetics	0.6438	0.5397	0.4736	0.4674	0.6242	**0.5497**

As shown in Tab. 3, ReCAST obtains a mean overall win rate of 0.572 and is preferred at four of the five budgets, most strongly at $(1,1,1,1)$ and $(2,1,1,1)$ (0.654 and 0.633), where both faithfulness and aesthetics improve. App. G shows paired generations behind these judgments and their selection protocol.

4.5 ABLATION STUDY: THE SINKHORN PROJECTION

We ablate the Sinkhorn projection of Sec. 3.4, which enforces a uniform total budget at each step, by training *no-Sink Rényi* variants that keep the adaptive kernel but omit the column constraint: $W^\star$ becomes the row-normalized kernel $W_{i,t} = \lambda_i K_{i,t} / \sum_s K_{i,s}$. Every reward still spends exactly its budget λ_i and the loss still consumes $T \cdot W_{i,t}$, but $\sum_i W_{i,t}$ is free, so each reward rides its own gain shape. All other training choices match the OCR experiment.

Table 4: Aggregate training-reward score for each $\boldsymbol{\lambda} = (\text{clip}, \text{hps}, \text{pick}, \text{ocr})$ budget in the training stage-OCR experiment, comparing static weighting, the adaptive kernel alone (no-Sink Rényi), and the full method. Scores are on the held-out OCR dataset at the last checkpoint (step 180), single matched training seed.

$\boldsymbol{\lambda}$	$(1,1,1,1)$	$(1,1,1,2)$	$(1,1,2,1)$	$(1,2,1,1)$	$(2,1,1,1)$
static	2.3848	3.2476	3.2948	2.5597	2.6980
no-Sink Rényi	2.4242	3.3514	**3.3781**	2.6721	2.6969
ReCAST (Sinkhorn)	**2.4650**	**3.3564**	3.3564	**2.7960**	**2.7038**

As shown in Tab. 4, no-Sink Rényi beats static on four of five budgets, and the full method beats no-Sink Rényi on four of five. Averaged across budgets, the adaptive kernel accounts for $+0.068$ of ReCAST's total $+0.099$ gain over static, and the Sinkhorn projection for the remaining $+0.031$ (this seed's gap, not the three-seed mean of Tab. 1; both parts of the decomposition share the seed). Reward-specific timing therefore identifies useful optimization steps, while the shared column budget keeps the resulting weights from concentrating on a few of them.

5 CONCLUSION

Multi-reward diffusion fine-tuning must decide both *how much* each reward matters and *when* it should act. ReCAST separates these decisions: the budget $\boldsymbol{\lambda}$ fixes each reward's total contribution, and its Rényi discriminability curve distributes it across timesteps.

On SD3.5-Medium, this temporal reallocation improves the OCR setting's aggregate training objective across five budgets and three seeds, with all four training rewards increasing on average, and it transfers: every held-out judge improves over static, including on the GenEval setting where the two methods tie on the training rewards, and the pairwise LLM judge prefers ReCAST at four of five budgets. *When* a reward is applied can therefore matter as much as *how much* weight it receives. We discuss the limitations of our work in App. F.

REPRODUCIBILITY STATEMENT

To support reproducibility, Secs. 3–4.1 specify the objective, estimator, allocation algorithm, data splits, and evaluation protocol. App. A.1 states the modeling assumptions, App. A.2 and App. A.3 provide the proofs, and Apps. B.1–B.2 give the expanded estimator and stochastic conditional-rollout procedure. App. D reports the per-budget three-seed results, while Apps. E.1–E.3 identify the reward models and checkpoints, training and evaluation hyperparameters, compute costs, and complete LLM-judge protocol and prompt. App. F documents the proprietary-API dependency and the cost of regenerating the gain curves.

AI USE STATEMENT

In this work, we used generative AI tools for the following tasks with required disclosure: "Generate synthetic data sets" "support qualitative and thematic data analysis," and "assist in the writing of proofs." Specifically, GPT IMAGE 1.5 and NANO BANANA PRO generated the external image sets used to estimate gain curves, and the Gemini-based LLM-as-a-Judge adapted from the MMRB2 protocol produced the pairwise evaluation reported in Sec. 4.4. Generative AI assisted with writing the error analysis in App. A.3. Generative AI supported the qualitative example presentation in App. G.

We have not used generative AI tools for the following other tasks with required disclosure: "help develop theoretical models or conceptual frameworks," "formulate mathematical claims," "provide critical ingredients for proving mathematical claims," "propose or refine hypotheses," "design or provide feedback on research methodology or experiments," "implement methods," "clean and reformat dataset," and "interpret results." The remaining required-disclosure tasks, "assist with translation" are not applicable to this work.

Additionally, we used generative AI tools for the following tasks with recommended disclosure: "create or edit software code," "draft parts of a research paper," "summarize or analyse existing literature," "sourcing/searching for information," "edit a research paper to improve readability," and "identify relevant literature." GPT and Claude assisted with literature survey, retrieval, manuscript polishing, and drafting summaries of experimental conclusions supplied by the authors. Coding agents launched author-specified experiments and wrote author-specified code to aggregate and summarize experimental measurements.

We have reviewed all AI-assisted work. The authors developed the research ideas, theory, mathematical claims and proofs, methodology, and experimental design; implemented the proposed method; specified all experimental configurations and summary metrics; reviewed and tested the generated code; checked the experiment launches; verified the identified literature; and reviewed the judge and summary outputs. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

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

Jooyoung Choi, Jungbeom Lee, Chaehun Shin, Sungwon Kim, Hyunwoo Kim, and Sungroh Yoon. Perception prioritized training of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

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

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, volume 26, 2013.

Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. Safe RLHF: Safe reinforcement learning from human feedback. *arXiv preprint arXiv:2310.12773*, 2023.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

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.

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

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

Tiankai Hang, Shuyang Gu, Chen Li, Jianmin Bao, Dong Chen, Han Hu, Xin Geng, and Baining Guo. Efficient diffusion training via Min-SNR weighting strategy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

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

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

Yushi Hu, Reyhane Askari-Hemmat, Melissa Hall, Emily Dinan, Luke Zettlemoyer, and Marjan Ghazvininejad. Multimodal RewardBench 2: Evaluating omni reward models for interleaved text and image. *arXiv preprint arXiv:2512.16899*, 2025.

Joel Jang, Seungone Kim, Bill Yuchen Lin, Yizhong Wang, Jack Hessel, Luke Zettlemoyer, Hannaneh Hajishirzi, Yejin Choi, and Prithviraj Ammanabrolu. Personalized soups: Personalized large language model alignment via post-hoc parameter merging. *arXiv preprint arXiv:2310.11564*, 2023.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. VinePPO: Refining credit assignment in RL training of LLMs. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 29557–29590. PMLR, 2025. URL `https://proceedings.mlr.press/v267/kazemnejad25a.html`.

Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in Neural Information Processing Systems*, 34:21696–21707, 2021.

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.

Yuhang Lai, Siyuan Wang, Shujun Liu, Xuanjing Huang, and Zhongyu Wei. ALaRM: Align language models via hierarchical rewards modeling. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 7817–7831, 2024.

Junzhe Li, Yutao Cui, Tao Huang, Weijie Kong, Yiming Cheng, Chuxuan Zeng, Yinping Ma, Chun Fan, Miles Yang, Zhao Zhong, and Liefeng Bo. MixGRPO: Unlocking flow-based GRPO efficiency with mixed ODE-SDE. *arXiv preprint arXiv:2507.21802*, 2025.

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

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2024.

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

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-GRPO: Training flow matching models via online RL. *arXiv preprint arXiv:2505.05470*, 2025.

Shih-Yang Liu, Xin Dong, Ximing Lu, Shizhe Diao, Peter Belcak, Mingjie Liu, Min-Hung Chen, Hongxu Yin, Yu-Chiang Frank Wang, Kwang-Ting Cheng, Yejin Choi, Jan Kautz, and Pavlo Molchanov. GDPO: Group reward-decoupled normalization policy optimization for multi-reward RL optimization. *arXiv preprint arXiv:2601.05242*, 2026.

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

Yuhang Ma, Yunhao Shui, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. HPSv3: Towards wide-spectrum human preference score. *arXiv preprint arXiv:2508.03789*, 2025.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, 2021.

Mihir Prabhudesai, Russell Mendonca, Zheyang Qin, Katerina Fragkiadaki, and Deepak Pathak. Video diffusion alignment via reward gradients. *arXiv preprint arXiv:2407.08737*, 2024.

Alexandre Ramé, Guillaume Couairon, Corentin Dancette, Jean-Baptiste Gaya, Mustafa Shukor, Laure Soulier, and Matthieu Cord. Rewarded soups: towards Pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Alfréd Rényi. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, pp. 547–561. University of California Press, 1961.

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

Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967.

Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward gaming. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. In *Advances in Neural Information Processing Systems*, volume 34, pp. 1415–1428, 2021.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process- and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8228–8238, 2024.

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

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.

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*, 2025.

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

Huaisheng Zhu, Teng Xiao, and Vasant G Honavar. DSPO: Direct score preference optimization for diffusion model alignment. In *The Thirteenth International Conference on Learning Representations*, 2025.

A SUPPLEMENTARY THEORY

In App. A.1, we provide notations and basic properties on rectified flow matching and DiffusionNFT. In App. A.2, we prove Theorem 3.1. In App. A.3, we bound the error induced by replacing each reward-specific positive policy π_i^+ with the common surrogate π^+, and show that it is $O(\sqrt{\mathrm{KL}(\pi^+\|\pi_i^+)})$ on the discriminability gain and propagates to the Sinkhorn weights.

A.1 BACKGROUND: RECTIFIED FLOW MATCHING AND DIFFUSIONNFT

Rectified flow matching. All models in this paper are rectified-flow generators (Esser et al., 2024; Lipman et al., 2022; Liu et al., 2022). A clean sample $\boldsymbol{x}_0 \sim \pi(\cdot \mid \boldsymbol{c})$ and noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$ are connected by the straight interpolation

$$\boldsymbol{x}_t \;=\; \alpha_t\, \boldsymbol{x}_0 + \sigma_t\, \boldsymbol{\epsilon}, \qquad \alpha_t = 1 - \sigma_t, \qquad \sigma_0 = 0, \quad \sigma_T = 1, \tag{20}$$

so $t{=}0$ is clean data and $t{=}T$ is pure noise. Here α_t is the interpolation coefficient, distinct from the Rényi order α of Sec. 3.2 and from the mixture coefficient $\alpha(\boldsymbol{x}_t)$ defined below. Differentiating Eq. (20) gives the target velocity $\boldsymbol{v} = \dot{\alpha}_t\, \boldsymbol{x}_0 + \dot{\sigma}_t\, \boldsymbol{\epsilon}$, which for the linear path is simply $\boldsymbol{v} = \boldsymbol{\epsilon} - \boldsymbol{x}_0$. A velocity model $\boldsymbol{v}_\theta$ is fit by regression on this target,

$$\mathcal{L}_{\mathrm{FM}}(\theta) \;=\; \mathbb{E}_{\boldsymbol{c},\, \boldsymbol{x}_0,\, \boldsymbol{\epsilon},\, t}\Big[\, \big\|\boldsymbol{v}_\theta(\boldsymbol{x}_t, \boldsymbol{c}, t) - \boldsymbol{v}\big\|_2^2 \,\Big], \tag{21}$$

and sampling integrates $\mathrm{d}\boldsymbol{x} = \boldsymbol{v}_\theta(\boldsymbol{x}_t, \boldsymbol{c}, t)\, \mathrm{d}\sigma_t$ backward from $t{=}T$ to $t{=}0$ over a discrete schedule $\{\sigma_t\}_{t=0}^{T}$ of T steps. Two features of Eq. (21) matter for what follows: the expectation over t makes every timestep a separate regression problem, so a per-timestep coefficient can be attached to each without changing the estimator; and $\boldsymbol{v}$ depends on the data only through $\boldsymbol{x}_0$, so tilting the data distribution by a reward moves the target while leaving the forward path Eq. (20) untouched. The latter is what lets Theorem 3.1 apply to the reward-tilted marginals in Sec. 3.2.

DiffusionNFT. We briefly review the DiffusionNFT framework (Zheng et al., 2025), which forms the basis of our multi-reward training objective. Theorems A.1 and A.2 below are from DiffusionNFT (Zheng et al., 2025); we restate them here, without proof, for the notation our objective builds on.

Positive Policy. Let $\pi^{\mathrm{old}}(\boldsymbol{x}_0 \mid \boldsymbol{c})$ denote the current, or old, policy and $r(\boldsymbol{x}_0, \boldsymbol{c}) := p(\boldsymbol{o} = 1 \mid \boldsymbol{x}_0, \boldsymbol{c})$ the optimality probability. The *positive policy* is defined as the old policy conditioned on optimality:

$$\pi^+(\boldsymbol{x}_0 \mid \boldsymbol{c}) \;:=\; \pi^{\mathrm{old}}(\boldsymbol{x}_0 \mid \boldsymbol{o} = 1, \boldsymbol{c}) \;=\; \frac{r(\boldsymbol{x}_0, \boldsymbol{c})}{p_{\pi^{\mathrm{old}}}(\boldsymbol{o} = 1 \mid \boldsymbol{c})}\, \pi^{\mathrm{old}}(\boldsymbol{x}_0 \mid \boldsymbol{c}). \tag{22}$$

With infinitely many samples from π^{old}, π^+ is simply the distribution of the positive subset: the old policy restricted to its "optimal" outputs.

The Coefficient $\alpha(\boldsymbol{x}_t)$. A key quantity linking the diffused positive marginal to the diffused old marginal is the scalar coefficient $\alpha(\boldsymbol{x}_t) \in [0, 1]$:

$$\alpha(\boldsymbol{x}_t) \;:=\; \frac{\pi_t^+(\boldsymbol{x}_t \mid \boldsymbol{c})}{\pi_t^{\mathrm{old}}(\boldsymbol{x}_t \mid \boldsymbol{c})} \cdot \mathbb{E}_{\pi^{\mathrm{old}}(\boldsymbol{x}_0 \mid \boldsymbol{c})}[r(\boldsymbol{x}_0, \boldsymbol{c})]. \tag{23}$$

It comes out of a posterior decomposition: the old posterior splits into a mixture of the positive and negative posteriors, weighted by $\alpha(\boldsymbol{x}_t)$ and $1 - \alpha(\boldsymbol{x}_t)$.

Theorem A.1 (Improvement direction, (Zheng et al., 2025))**.** *Let $\boldsymbol{v}^+$, $\boldsymbol{v}^-$, and $\boldsymbol{v}^{\mathrm{old}}$ be the velocity models for the policies π^+, π^-, and π^{old}, respectively. The directional differences between these models are proportional:*

$$\Delta \;:=\; \big[1 - \alpha(\boldsymbol{x}_t)\big]\, \big[\boldsymbol{v}^{\mathrm{old}}(\boldsymbol{x}_t, \boldsymbol{c}, t) - \boldsymbol{v}^-(\boldsymbol{x}_t, \boldsymbol{c}, t)\big] \;=\; \alpha(\boldsymbol{x}_t)\, \big[\boldsymbol{v}^+(\boldsymbol{x}_t, \boldsymbol{c}, t) - \boldsymbol{v}^{\mathrm{old}}(\boldsymbol{x}_t, \boldsymbol{c}, t)\big], \tag{24}$$

where $0 \le \alpha(\boldsymbol{x}_t) \le 1$ is defined in Eq. (23)*.*

So Δ is an improvement direction in velocity space: moving from $\boldsymbol{v}^{\mathrm{old}}$ toward $\boldsymbol{v}^+$ is the same direction as moving away from $\boldsymbol{v}^-$.

Theorem A.2 (Policy optimization, (Zheng et al., 2025)). *Consider the training objective*

$$\mathcal{L}(\theta) \;=\; \mathbb{E}_{\boldsymbol{c},\,\pi^{\text{old}}(\boldsymbol{x}_0|\boldsymbol{c}),\,\boldsymbol{\epsilon},\,t}\Big[r\,\|\boldsymbol{v}_\theta^+(\boldsymbol{x}_t,\boldsymbol{c},t)-\boldsymbol{v}\|_2^2 \;+\; (1-r)\,\|\boldsymbol{v}_\theta^-(\boldsymbol{x}_t,\boldsymbol{c},t)-\boldsymbol{v}\|_2^2\Big], \tag{25}$$

where $\boldsymbol{v}=\dot\alpha_t\,\boldsymbol{x}_0+\dot\sigma_t\,\boldsymbol{\epsilon}$ is the target velocity from the forward process, and the implicit positive and negative policies are parameterized as

$$\boldsymbol{v}_\theta^+(\boldsymbol{x}_t,\boldsymbol{c},t) \;:=\; (1-\beta)\,\boldsymbol{v}^{\text{old}}(\boldsymbol{x}_t,\boldsymbol{c},t)+\beta\,\boldsymbol{v}_\theta(\boldsymbol{x}_t,\boldsymbol{c},t), \tag{26}$$

$$\boldsymbol{v}_\theta^-(\boldsymbol{x}_t,\boldsymbol{c},t) \;:=\; (1+\beta)\,\boldsymbol{v}^{\text{old}}(\boldsymbol{x}_t,\boldsymbol{c},t)-\beta\,\boldsymbol{v}_\theta(\boldsymbol{x}_t,\boldsymbol{c},t). \tag{27}$$

Given unlimited data and model capacity, the optimal solution satisfies

$$\boldsymbol{v}_{\theta^*}(\boldsymbol{x}_t,\boldsymbol{c},t) \;=\; \boldsymbol{v}^{\text{old}}(\boldsymbol{x}_t,\boldsymbol{c},t) \;+\; \frac{2}{\beta}\,\Delta(\boldsymbol{x}_t,\boldsymbol{c},t), \tag{28}$$

where Δ is defined in Eq. (24).

The optimum therefore sits along the reinforcement-guidance direction Δ, at strength $2/\beta$ relative to $\boldsymbol{v}^{\text{old}}$, so the policy improves without any likelihood being estimated.

A.2 Proof of Theorem 3.1

Proof. Let $\rho := \pi^a/\pi^b$ and $M_\alpha := \mathbb{E}_{\pi^b}[\rho^\alpha] = \int \rho^\alpha\,\pi^b\,\mathrm{d}\boldsymbol{x} = \int (\pi^a)^\alpha\,(\pi^b)^{1-\alpha}\,\mathrm{d}\boldsymbol{x}$. By definition, $D_\alpha = \frac{1}{\alpha-1}\log M_\alpha$, so

$$\frac{\mathrm{d}}{\mathrm{d}t}D_\alpha \;=\; \frac{1}{\alpha-1}\cdot\frac{\dot M_\alpha}{M_\alpha}.$$

It suffices to show $\dot M_\alpha = -\frac{g^2}{2}\,\alpha(\alpha-1)\int \rho^\alpha\|\nabla\log\rho\|^2\,\pi^b\,\mathrm{d}\boldsymbol{x}$.

Step 1: Differentiating M_α. Since $\rho=\pi^a/\pi^b$, we have $\partial_t\rho = (\partial_t\pi^a)/\pi^b - \rho\,(\partial_t\pi^b)/\pi^b$, so

$$\dot M_\alpha \;=\; \int\big[\alpha\,\rho^{\alpha-1}\,(\partial_t\rho)\,\pi^b+\rho^\alpha\,\partial_t\pi^b\big]\,\mathrm{d}\boldsymbol{x} \;=\; \alpha\int\rho^{\alpha-1}\,\partial_t\pi^a\,\mathrm{d}\boldsymbol{x} \;-\; (\alpha{-}1)\int\rho^\alpha\,\partial_t\pi^b\,\mathrm{d}\boldsymbol{x}.$$

Step 2: Substituting the Fokker–Planck equation. Both π^a and π^b satisfy $\partial_t\pi = -\nabla\cdot(\boldsymbol{u}\pi)+\frac{g^2}{2}\Delta\pi$, so

$$\dot M_\alpha \;=\; \underbrace{\alpha\int\rho^{\alpha-1}\big[-\nabla\cdot(\boldsymbol{u}\pi^a)+\tfrac{g^2}{2}\Delta\pi^a\big]\,\mathrm{d}\boldsymbol{x}}_{=:\,I_a} \;-\; \underbrace{(\alpha{-}1)\int\rho^\alpha\big[-\nabla\cdot(\boldsymbol{u}\pi^b)+\tfrac{g^2}{2}\Delta\pi^b\big]\,\mathrm{d}\boldsymbol{x}}_{=:\,I_b}.$$

Step 3: Drift terms cancel. For the drift part of I_a, integrate by parts (boundary terms vanish by decay at infinity):

$$-\alpha\int\rho^{\alpha-1}\,\nabla\cdot(\boldsymbol{u}\pi^a)\,\mathrm{d}\boldsymbol{x} \;=\; \alpha\int\boldsymbol{u}\pi^a\cdot\nabla(\rho^{\alpha-1})\,\mathrm{d}\boldsymbol{x} \;=\; \alpha(\alpha{-}1)\int\boldsymbol{u}\cdot\nabla\rho\;\rho^{\alpha-2}\pi^a\,\mathrm{d}\boldsymbol{x}.$$

Using $\pi^a=\rho\pi^b$, this equals $\alpha(\alpha{-}1)\int\boldsymbol{u}\cdot\nabla\rho\;\rho^{\alpha-1}\pi^b\,\mathrm{d}\boldsymbol{x}$. For the drift part of I_b:

$$(\alpha{-}1)\int\rho^\alpha\,\nabla\cdot(\boldsymbol{u}\pi^b)\,\mathrm{d}\boldsymbol{x} \;=\; -(\alpha{-}1)\int\boldsymbol{u}\pi^b\cdot\nabla(\rho^\alpha)\,\mathrm{d}\boldsymbol{x} \;=\; -\alpha(\alpha{-}1)\int\boldsymbol{u}\cdot\nabla\rho\;\rho^{\alpha-1}\pi^b\,\mathrm{d}\boldsymbol{x}.$$

Hence the drift contributions from I_a and I_b sum to zero.

Step 4: Diffusion terms via integration by parts. It remains to evaluate the diffusive parts. For the I_a diffusion term, apply Green's first identity (integration by parts twice):

$$\begin{aligned}\tfrac{g^2}{2}\,\alpha\int\rho^{\alpha-1}\,\Delta\pi^a\,\mathrm{d}\boldsymbol{x} \;&=\; -\tfrac{g^2}{2}\,\alpha\int\nabla(\rho^{\alpha-1})\cdot\nabla\pi^a\,\mathrm{d}\boldsymbol{x}\\ &=\; -\tfrac{g^2}{2}\,\alpha(\alpha{-}1)\int\rho^{\alpha-2}\,\nabla\rho\cdot\nabla\pi^a\,\mathrm{d}\boldsymbol{x}.\end{aligned}$$

Now use $\nabla\pi^a = \nabla(\rho\pi^b) = \pi^b\nabla\rho + \rho\nabla\pi^b$:

$$= -\tfrac{g^2}{2}\,\alpha(\alpha{-}1)\int \rho^{\alpha-2}\big[\pi^b\|\nabla\rho\|^2 + \rho\,\nabla\rho{\cdot}\nabla\pi^b\big]\,\mathrm{d}\boldsymbol{x}.$$

For the I_b diffusion term:

$$-\tfrac{g^2}{2}\,(\alpha{-}1)\int \rho^\alpha\,\Delta\pi^b\,\mathrm{d}\boldsymbol{x} \;=\; \tfrac{g^2}{2}\,(\alpha{-}1)\int \nabla(\rho^\alpha){\cdot}\nabla\pi^b\,\mathrm{d}\boldsymbol{x}$$
$$= \tfrac{g^2}{2}\,\alpha(\alpha{-}1)\int \rho^{\alpha-1}\,\nabla\rho{\cdot}\nabla\pi^b\,\mathrm{d}\boldsymbol{x}.$$

Adding the two diffusive contributions, the $\nabla\rho{\cdot}\nabla\pi^b$ terms cancel:

$$-\tfrac{g^2}{2}\,\alpha(\alpha{-}1)\int \rho^{\alpha-2}\,\rho\,\nabla\rho{\cdot}\nabla\pi^b\,\mathrm{d}\boldsymbol{x} \;+\; \tfrac{g^2}{2}\,\alpha(\alpha{-}1)\int \rho^{\alpha-1}\,\nabla\rho{\cdot}\nabla\pi^b\,\mathrm{d}\boldsymbol{x} \;=\; 0,$$

and we are left with

$$\dot{M}_\alpha \;=\; -\tfrac{g(t)^2}{2}\,\alpha(\alpha{-}1)\int \rho^{\alpha-2}\,\|\nabla\rho\|^2\,\pi^b\,\mathrm{d}\boldsymbol{x}.$$

Step 5: Rewrite in terms of $\nabla\log\rho$. Since $\nabla\rho = \rho\,\nabla\log\rho$, we have $\rho^{\alpha-2}\|\nabla\rho\|^2 = \rho^\alpha\|\nabla\log\rho\|^2$, giving

$$\dot{M}_\alpha \;=\; -\tfrac{g(t)^2}{2}\,\alpha(\alpha{-}1)\int \rho^\alpha\,\|\nabla\log\rho\|^2\,\pi^b\,\mathrm{d}\boldsymbol{x}.$$

Since $\frac{\mathrm{d}}{\mathrm{d}t}D_\alpha = \frac{1}{\alpha-1}\cdot\frac{\dot{M}_\alpha}{M_\alpha}$, the factor $(\alpha{-}1)$ cancels, giving $\frac{\mathrm{d}}{\mathrm{d}t}D_\alpha = -\frac{g(t)^2}{2}\cdot\alpha\cdot\frac{\int\rho^\alpha\|\nabla\log\rho\|^2\pi^b\,\mathrm{d}\boldsymbol{x}}{\int\rho^\alpha\pi^b\,\mathrm{d}\boldsymbol{x}} = -\frac{g(t)^2}{2}\cdot\alpha\cdot\mathbb{E}_{\pi^{(\alpha)}}[\|\nabla\log\rho\|^2]$, which is Eq. (10). □

A.3 Error induced by the surrogate

We quantify the error introduced by replacing the reward-specific positive policy π_i^+ with a common external positive policy π^+ when estimating the reward-specific temporal weights. The main result shows that this approximation is stable when π^+ is close to π_i^+ in KL divergence. Throughout this subsection we fix a prompt $\boldsymbol{c}$, suppress it from the notation, and take $\alpha > 1$ as in the main text.

Bounding the discriminability error. Assume the reward is bounded, $0 \le r_i(\boldsymbol{x}_0) \le R_i$ with $Z_i > 0$, so that $0 \le \rho_{i,t}(\boldsymbol{x}_t) \le M_i := R_i/Z_i$ for every t, and define the policy mismatch $\epsilon_i := \mathrm{KL}\big(\pi^+\,\|\,\pi_i^+\big)$. Because π_t^+ and $\pi_{i,t}^+$ are obtained by applying the same diffusion kernel, the data-processing inequality gives $\mathrm{KL}\big(\pi_t^+\,\|\,\pi_{i,t}^+\big) \le \epsilon_i$, and Pinsker's inequality then gives $\mathrm{TV}\big(\pi_t^+,\pi_{i,t}^+\big) \le \sqrt{\epsilon_i/2}$. Since $f_{i,t} := \rho_{i,t}^{\alpha-1}$ takes values in $[0, M_i^{\alpha-1}]$,

$$\Big|\mathbb{E}_{\pi_t^+}[f_{i,t}] - \mathbb{E}_{\pi_{i,t}^+}[f_{i,t}]\Big| \;\le\; M_i^{\alpha-1}\,\mathrm{TV}\big(\pi_t^+,\pi_{i,t}^+\big) \;\le\; M_i^{\alpha-1}\sqrt{\frac{\epsilon_i}{2}} \;=:\; \eta_i. \tag{29}$$

Moreover $\mathbb{E}_{\pi_{i,t}^+}[f_{i,t}] = \exp\big((\alpha-1)D_{i,t}\big) \ge 1$. Write $\widetilde{D}_{i,t} := \frac{1}{\alpha-1}\log\mathbb{E}_{\pi_t^+}[f_{i,t}]$ for the surrogate log-moment, whose differences give the surrogate gain of Eq. (12). Whenever $\eta_i < 1$, Eq. (29) yields

$$\big|\widetilde{D}_{i,t} - D_{i,t}\big| \;\le\; b_i := \frac{-\log(1-\eta_i)}{\alpha-1} \;=\; \frac{M_i^{\alpha-1}}{\alpha-1}\sqrt{\frac{\epsilon_i}{2}} + O(\epsilon_i), \tag{30}$$

so the surrogate divergence is accurate to $O\big(\sqrt{\epsilon_i}\big)$.

Error in the per-timestep gain. Applying Eq. (30) at the two adjacent timesteps of the gain $\Delta D_{i,t} = D_{i,t-1} - D_{i,t}$ and of its surrogate $\widetilde{\Delta D}_{i,t} = \widetilde{D}_{i,t-1} - \widetilde{D}_{i,t}$ gives

$$\big|\widetilde{\Delta D}_{i,t} - \Delta D_{i,t}\big| \;\le\; 2b_i \;\lesssim\; \frac{\sqrt{2}\,M_i^{\alpha-1}}{\alpha-1}\sqrt{\epsilon_i}, \tag{31}$$

where the final comparison holds in the small-mismatch regime.

Error after normalizing the temporal profile. The allocation uses the mean-normalized gain of Eq. (13), $\overline{\Delta D}_{i,t} = T\,\Delta D_{i,t}/G_i$ with $G_i := \sum_t \Delta D_{i,t}$, and its surrogate $\widetilde{\overline{\Delta D}}_{i,t} := T\,\widetilde{\Delta D}_{i,t}/\widetilde{G}_i$. Because the gains telescope, $G_i = D_{i,0} - D_{i,T}$ and $\widetilde{G}_i = \widetilde{D}_{i,0} - \widetilde{D}_{i,T}$, so $|\widetilde{G}_i - G_i| \le 2b_i$. If $G_i > 2b_i$, then for every t

$$\left|\widetilde{\overline{\Delta D}}_{i,t} - \overline{\Delta D}_{i,t}\right| \le T\,\frac{|\widetilde{\Delta D}_{i,t} - \Delta D_{i,t}|}{\widetilde{G}_i} + \overline{\Delta D}_{i,t}\,\frac{|G_i - \widetilde{G}_i|}{\widetilde{G}_i} \le \frac{4T\,b_i}{G_i - 2b_i} = O\!\left(\frac{T\,M_i^{\alpha-1}}{(\alpha-1)\,G_i}\sqrt{\epsilon_i}\right). \tag{32}$$

Propagation to Sinkhorn weights. Finally, consider the entropy-regularized allocation, written with the cost notation of Eq. (17),

$$\boldsymbol{W}(\boldsymbol{C}) = \operatorname*{arg\,min}_{\boldsymbol{W}\in\mathcal{U}(\boldsymbol{\lambda},\boldsymbol{\nu})} \sum_{i,t}\Big[\,C_{i,t}\,W_{i,t} + \tau\,W_{i,t}\log W_{i,t}\,\Big], \qquad C_{i,t} := -\overline{\Delta D}_{i,t}, \tag{33}$$

where $\mathcal{U}(\boldsymbol{\lambda},\boldsymbol{\nu})$ fixes the reward marginals to $\boldsymbol{\lambda}$ and the timestep marginals to $\boldsymbol{\nu}$, $\tau > 0$ is the Sinkhorn temperature, and $B := \sum_i \lambda_i = \sum_t \nu_t$ is the total transported mass; at $\nu_t \equiv 1/T$ and $\tau = 1$, Eq. (33) is exactly Eq. (17), with feasible set $\mathcal{U}(\boldsymbol{\lambda})$ of Eq. (15) and solution $\boldsymbol{W}^\star$. Let $\boldsymbol{W}^\star := \boldsymbol{W}(\boldsymbol{C})$ and $\widetilde{\boldsymbol{W}} := \boldsymbol{W}(\widetilde{\boldsymbol{C}})$, where $\widetilde{\boldsymbol{C}} := \big(-\widetilde{\overline{\Delta D}}_{i,t}\big)_{i,t}$ is the cost built from the surrogate gains. The entropic term is τ/B-strongly convex with respect to the ℓ_1 norm on measures of mass B, so the optimality conditions for Eq. (33) at $\boldsymbol{W}^\star$ and $\widetilde{\boldsymbol{W}}$ give $\frac{\tau}{B}\|\widetilde{\boldsymbol{W}} - \boldsymbol{W}^\star\|_1^2 \le \langle \boldsymbol{C} - \widetilde{\boldsymbol{C}},\, \widetilde{\boldsymbol{W}} - \boldsymbol{W}^\star\rangle$, and Hölder's inequality then yields the stability estimate $\|\widetilde{\boldsymbol{W}} - \boldsymbol{W}^\star\|_1 \le \frac{B}{\tau}\|\widetilde{\boldsymbol{C}} - \boldsymbol{C}\|_\infty$. Since $\|\widetilde{\boldsymbol{C}} - \boldsymbol{C}\|_\infty$ is bounded by Eq. (32), the end-to-end bound for sufficiently small policy mismatch is

$$\|\widetilde{\boldsymbol{W}} - \boldsymbol{W}^\star\|_1 \;\le\; \frac{4BT}{\tau}\max_i \frac{b_i}{G_i - 2b_i} \;=\; O\!\left(\frac{BT}{\tau}\max_i \frac{M_i^{\alpha-1}}{(\alpha-1)\,G_i}\sqrt{\mathrm{KL}\big(\pi^+\|\pi_i^+\big)}\right). \tag{34}$$

Eq. (34) gives a direct interpretation of the approximation: a shared positive policy produces similar temporal weights whenever it is close to the reward-specific positive policy, with the square root coming from Pinsker's inequality. Rewards with larger total discriminability G_i are less sensitive to the approximation, while a smaller Sinkhorn temperature τ makes the final allocation more sensitive to errors in the estimated temporal profile.

B Gain estimation: algorithm, α-sensitivity, and curve details

Throughout this section, we assume that the diffused external-generator marginal π_t^+ approximates every reward-specific positive marginal $\pi_{i,t}^+$.

B.1 Algorithm

Alg. 2 writes out Alg. 1 in full, separated into its three phases, and implements the estimator of Sec. 3.3. Phase 1 forward-noises clean π^+ samples to obtain $\boldsymbol{x}_t \sim \pi_t^+$; Phase 2 rolls out under π^{old} to estimate the unnormalized conditional reward at each $\boldsymbol{x}_t$; Phase 3 differences the resulting log-moments, at which point the marginal $\mathbb{E}^{\mathrm{old}}[r_i \mid \boldsymbol{c}]$ drops out and never has to be formed, leaving the surrogate gain of Eq. (12).

B.2 Stochastic conditional rollouts

Although the background in App. A.1 introduces the deterministic probability-flow ODE, the conditional rollouts used by Alg. 1 are generated with a stochastic reverse-SDE solver. For each forward-corrupted state $\boldsymbol{x}_t$, we replicate the same latent K times and integrate the old policy from t to 0. At every reverse step, an independent Gaussian increment is drawn for each replica and an Euler–Maruyama update is applied; we use stochasticity $\eta = 0.7$ in all experiments. Consequently, a fixed $\boldsymbol{x}_t$ produces K non-degenerate terminal samples, whose rewards are averaged to form a Monte Carlo estimate of $\mu_{i,t}(\boldsymbol{x}_t)$. We interpret these samples as draws from the old model's reverse-SDE conditional completion distribution, which approximates $\pi^{\mathrm{old}}(\boldsymbol{x}_0 \mid \boldsymbol{x}_t, \boldsymbol{c})$ up to model and time-discretization error.

Algorithm 2 Estimation of per-step Rényi discriminability gain proxies $\widehat{\widetilde{\Delta D}}_{i,t}$ from π^+ (the expanded form of Alg. 1)

Require: Prompt $\boldsymbol{c}$, current policy π^{old}, external high-reward generator approximating $\pi^+(\boldsymbol{x}_0 \mid \boldsymbol{c})$, rewards $r_1, \ldots, r_m$, Rényi order $\alpha > 0$ with $\alpha \neq 1$, number of π^+ samples N, number of rollouts K, noise schedule $\{\sigma_t\}_{t=0}^{T}$

Ensure: Per-step gain proxies $\widehat{\widetilde{\Delta D}}_{i,t}$ for each reward i and timestep t

Phase 1: Sample $\boldsymbol{x}_t \sim \pi_t^+$ by forward-noising clean π^+ samples
for $n = 1, \ldots, N$ **do**
  Draw clean sample $\widetilde{\boldsymbol{x}}_0^{(n)} \sim \pi^+(\cdot \mid \boldsymbol{c})$ from the stronger external generator
  for each timestep $t \in \{0, 1, \ldots, T\}$ **do**
    Sample $\boldsymbol{\epsilon}_t^{(n)} \sim \mathcal{N}(0, I)$ and set $\boldsymbol{x}_t^{(n)} \leftarrow (1 - \sigma_t)\, \widetilde{\boldsymbol{x}}_0^{(n)} + \sigma_t\, \boldsymbol{\epsilon}_t^{(n)}$ *// $\boldsymbol{x}_t^{(n)} \sim \pi_t^+$*
  end for
end for
Phase 2: Rollouts under π^{old} from each $\boldsymbol{x}_t^{(n)}$ to estimate the conditional reward
for $n = 1, \ldots, N$ **do**
  for each timestep $t \in \{0, 1, \ldots, T\}$ **do**
    if $t = 0$ **then** *// $\widetilde{\boldsymbol{x}}_0^{(n)}$ is already clean: no rollout, single deterministic evaluation*
      $\widetilde{\boldsymbol{x}}_0^{(n,0,1)} \leftarrow \widetilde{\boldsymbol{x}}_0^{(n)}$; evaluate $r_i(\widetilde{\boldsymbol{x}}_0^{(n,0,1)}, \boldsymbol{c})$ for all i
    else
      for $k = 1, \ldots, K$ **do**
        Run independent rollout from $\boldsymbol{x}_t^{(n)}$ to completion under π^{old}: $\boldsymbol{x}_t^{(n)} \to \boldsymbol{x}_0^{(n,t,k)}$
        Evaluate $r_i(\boldsymbol{x}_0^{(n,t,k)}, \boldsymbol{c})$ for all rewards i
      end for
    end if
  end for
end for
Phase 3: Per-step gains, with denominator cancellation
for each reward i **do**
  for each timestep $t \in \{0, 1, \ldots, T\}$ **do**
    for $n = 1, \ldots, N$ **do**
      $\hat{\mu}_{i,t}^{(n)} \leftarrow \frac{1}{K_t} \sum_{k=1}^{K_t} r_i(\boldsymbol{x}_0^{(n,t,k)}, \boldsymbol{c})$, with $K_t = 1$ if $t = 0$ else K *// Conditional reward at $\boldsymbol{x}_t^{(n)}$*
    end for
    $\widehat{\widetilde{D}}_{i,t} \leftarrow \frac{1}{\alpha - 1} \log\Big(\max\{10^{-10}, \frac{1}{N} \sum_{n=1}^{N} \big[\hat{\mu}_{i,t}^{(n)}\big]^{\alpha-1}\} \Big)$ *// Surrogate log-moment, with a numerical floor*
  end for
  for $t = 1, \ldots, T$ **do**
    $\widehat{\widetilde{\Delta D}}_{i,t} \leftarrow \big[\widehat{\widetilde{D}}_{i,t-1} - \widehat{\widetilde{D}}_{i,t}\big]_+$ *// Eq. (12), clipped as in Sec. 3.4*
  end for
end for

B.3 α-SENSITIVITY OF THE RÉNYI ESTIMATOR

To analyze the influence of α, we sweep $\alpha \in \{0.25, 0.5, 0.75, 1.5, 2, 4, 8\}$ and plot per-step gain, mean-normalized gain, and the Sinkhorn matrix they project to in Fig. 2. The sweep covers the five rewards estimated jointly on the shared T=10 grid. For $0 < \alpha < 1$ the exponent $(\alpha - 1)$ is negative, so the log-moment is dominated by the lower tail of $\hat{\mu}_{i,t}$ rather than the upper one. Of the two guarantees we rely on, only one needs $\alpha > 1$: Theorem 3.1 gives $\Delta D_{i,t} \geq 0$ for every $\alpha > 0$ with $\alpha \neq 1$, whereas D_α upper-bounds the KL only above 1. Below 1 the gain therefore keeps its monotonicity but loses its link to the original objective, and we report these orders only to show what the estimator does outside its intended range. Peak locations are stable through the middle of the range and move only at its ends: AESTHETIC peaks at the clean end for $\alpha \leq 0.75$ but near the noisy end for $\alpha \geq 1.5$, and at α=8 HPSV2 and PICKSCORE lose their own peaks closer to clean and collapse onto IMAGEREWARD's at t=9. Raising α also localizes the gains: IMAGEREWARD's normalized peak grows from 1.61× its own mean at α=0.25 to 2.12× at α=2 and 3.25× at α=8. The projected weights follow, and their excursions from uniform are mildest in the intermediate range, spanning 0.29 to 1.60 times $1/(mT)$ at α=2 against 0.24 to 1.89 at α=0.25 and 0.41 to 2.24 at α=8, which is why we default to $\alpha = 2$ in the main text.

B.4 π^+ VS. π^{old} SELF-ROLLOUT GAIN CURVES

In this section, we validate the design of π^+ sampling. To this end, we run the identical estimator and Sinkhorn recipe with π^{old} self-rollouts in place of π^+ samples, replacing the exponent $\alpha - 1$ by α so that both runs target the same $D_{i,t}$, since $\mathbb{E}_{\pi_t^{\text{old}}}[\rho_{i,t}(\boldsymbol{x}_t)^\alpha] = \mathbb{E}_{\pi_{i,t}^+}[\rho_{i,t}(\boldsymbol{x}_t)^{\alpha-1}]$ (Sec. 3.3), holding the training stage-OCR prompt set, the four training rewards (CLIPSCORE, HPSV2, OCR, PICKSCORE), α=2 and T=25 fixed, so the two runs differ only in the sampling distribution. As shown in Fig. 3, the π^+-sampled ΔD curves (top) are smooth and single-peaked, with the peak location differing by reward, while the π^{old}-sampled curves are jagged and non-monotonic step-to-step, with per-step gain repeatedly spiking and collapsing back to the zero clip. The leftmost column makes the same point quantitatively. At the clean end the π^{old} estimate gives $D_{i,t}$ of 0.016, 0.020, and 0.001 for CLIPSCORE, HPSV2, and PICKSCORE, against 0.245, 0.501, and 0.209 under π^+: a 15–200× collapse, and precisely the $\rho_{i,t} \approx 1$, $D_{i,t} \approx 0$ degeneracy anticipated in Sec. 3.3. OCR is the one reward that retains appreciable discriminability, but its ΔD curve is unstable in t. The Sinkhorn matrices built from these noisy curves (right column) are correspondingly noisy themselves, oscillating step-to-step rather than smoothly redistributing weight toward each reward's high-gain region. In short: π^+ sampling is what makes the estimated curve usable as a Sinkhorn kernel input.

The jaggedness is not an artifact of the budget. A natural objection is that the π^{old} curves are jagged only because the estimation budget is small. The bottom row of Fig. 3 rules this out: it recomputes the same π^{old} run at N=32 and K=64, a 4× increase in both the samples per prompt and the rollouts per $\boldsymbol{x}_t$ and hence 16× the rollout cost, so the two π^{old} rows differ in (N, K) alone. The ΔD curves stay just as jagged, so the shape the Sinkhorn kernel would consume is not yet stable. The extra budget therefore almost does not improve smoothness. π^+ sampling instead reaches a usable curve at $1/16$ of the rollout cost of this row.

B.5 SENSITIVITY TO THE π^+ SURROGATE GENERATOR

The curves above approximate π^+ by a single external generator, GPT IMAGE 1.5. To test how much the resulting weights depend on the specific π^+, we regenerate the training stage-OCR π^+ image set with an independent generator, NANO BANANA PRO (GEMINI 3 PRO IMAGE), same prompts and N=8 images per prompt, and rerun the identical estimator. Tab. 5 compares the two per-step curves reward-by-reward. The curves are highly correlated (Pearson $r \in [0.84, 0.92]$, Spearman $\rho \in [0.82, 0.92]$) and peak within 0–2 steps of each other. Fig. 4 shows the two curves side by side: the per-step gains, mean-normalized gains, and the Sinkhorn matrices they project to are visually near-interchangeable. The gain curve, and hence the Sinkhorn weights built from it, is therefore largely a property of the reward and the trajectory, not of which strong external generator stands in for π^+.

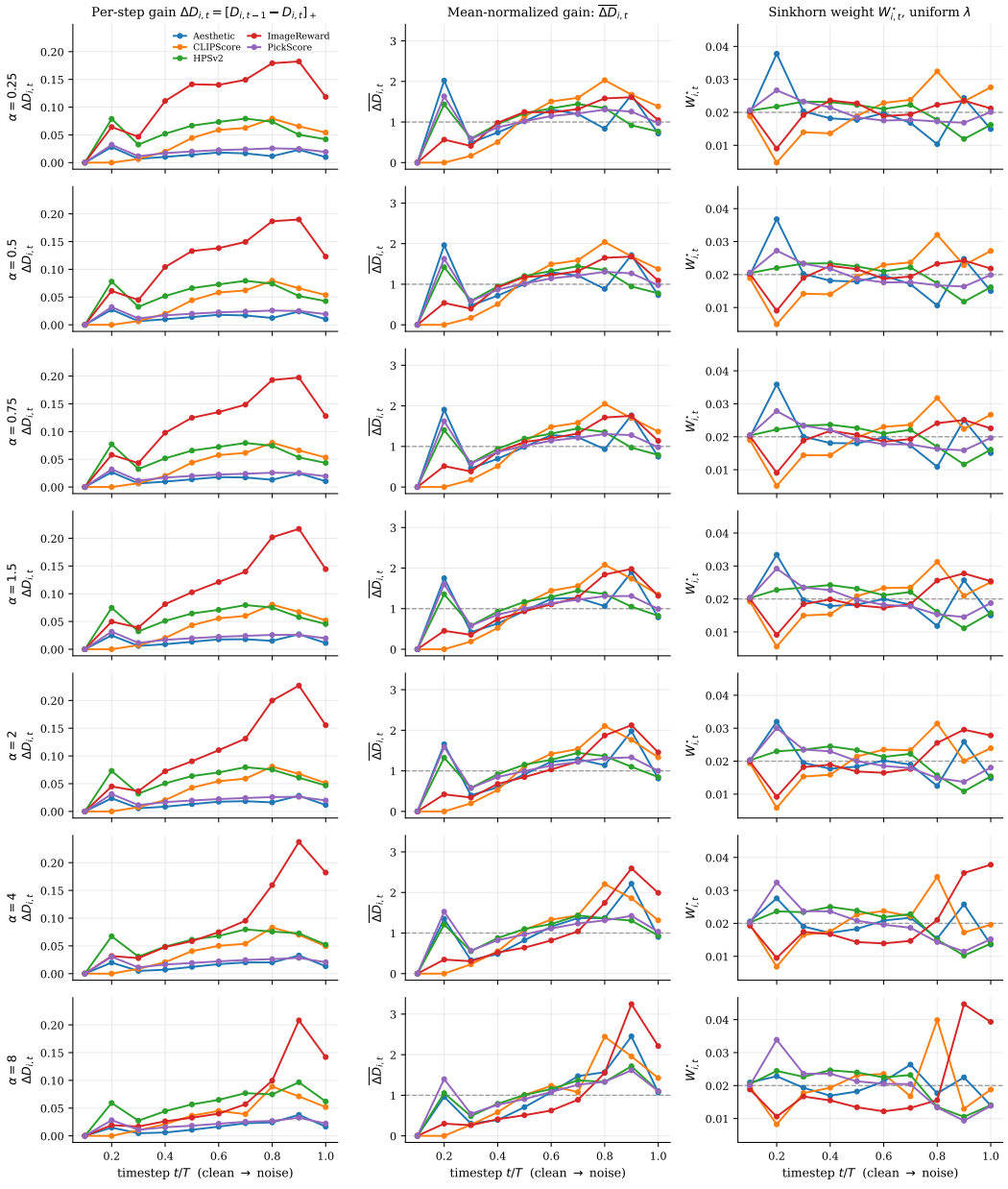


Figure 2: α-sensitivity of the π^{+}-sampling Rényi estimator on the SD3.5-Medium run of Sec. 3.5. **Rows:** $\alpha \in \{0.25, 0.5, 0.75, 1.5, 2, 4, 8\}$. **Columns:** the same three quantities as Fig. 1, per-step gain $\Delta D_{i,t}$ (left), mean-normalized gain $\overline{\Delta D}_{i,t}$ (Eq. (13), center), and the Sinkhorn matrix $W^{\star}_{i,t}$ (Eq. (18)) at uniform $\lambda_i{=}1/m$ (right).

Table 5: Surrogate sensitivity of the training stage-OCR gain curve: GPT IMAGE 1.5 vs. NANO BANANA PRO as the π^{+} generator, all other estimator settings identical. r and ρ are Pearson and Spearman correlations between the two per-step curves over the $T{=}25$ steps. "Peak t" gives the argmax timestep of $\Delta D_{i,t}$ in the paper's convention ($t{=}1$ clean, $t{=}T$ pure noise); "total gain" is the ratio of $\sum_t \Delta D_{i,t}$ under NANO BANANA PRO to that under GPT IMAGE 1.5.

Reward	Pearson r	Spearman ρ	Peak t	Total gain
CLIPSCORE	0.879	0.912	$17 \to 19$	$0.94\times$
HPSV2	0.847	0.829	$15 \to 16$	$0.81\times$
OCR	0.862	0.844	$17 \to 17$	$0.92\times$
PICKSCORE	0.919	0.915	$15 \to 16$	$0.84\times$

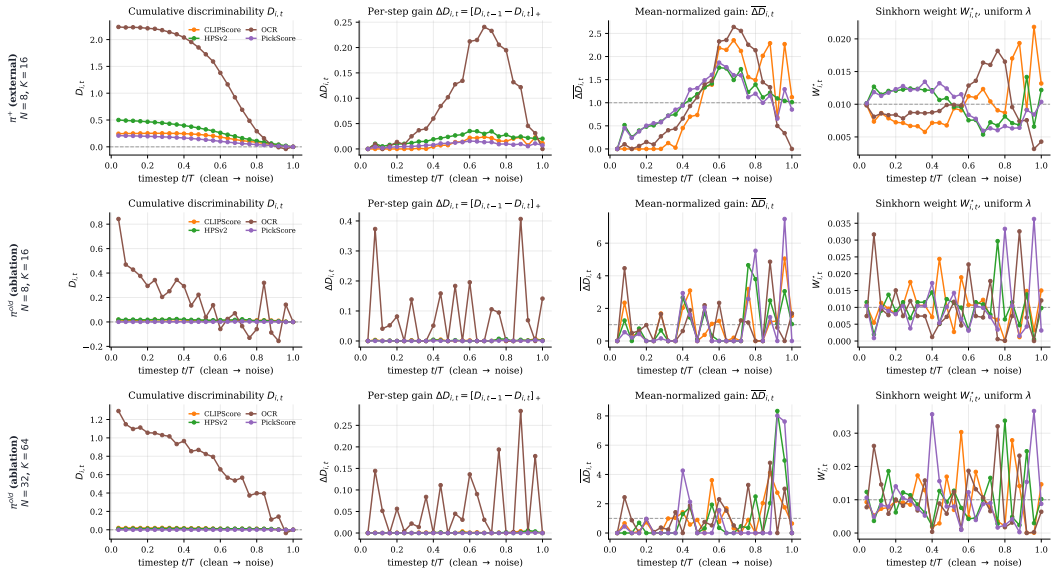


Figure 3: Cumulative discriminability $D_{i,t}$, per-step gain $\Delta D_{i,t}$, mean-normalized gain $\overline{\Delta D}_{i,t}$, and Sinkhorn matrix $W^\star_{i,t}$ at uniform $\lambda_i{=}1/m$, left to right in the order they are derived, under the two sampling distributions. The leftmost column is the estimator's own $D_{i,t}$, shifted by the single t-independent constant that sets $D_{i,T}{=}0$ at pure noise, where $\pi^+_{i,T}{=}\pi^{\text{old}}_T$; it is not a cumulative sum of the clipped gains beside it. Under π^{old} almost all of the estimated discriminability sits on OCR, the other three rewards coming out nearly indiscriminable at every step, and the OCR curve is not even monotone in t: it dips below zero, which Theorem 3.1 rules out in population. **Top:** π^+ samples from GPT IMAGE 1.5, at the $N{=}8$, $K{=}16$ budget used throughout the paper. **Middle:** π^{old} self-rollouts at that same budget. **Bottom:** the same π^{old} run at the full $N{=}32$, $K{=}64$ estimation budget, $16\times$ the rollout cost. All three rows are matched estimator runs on the training stage-OCR curve: same four training rewards (CLIPSCORE, HPSV2, OCR, PICKSCORE), same prompt set, same $\alpha{=}2$ and $T{=}25$; the top two differ only in the sampling distribution, the bottom two only in (N, K). Both π^{old} rows are visibly noisier at every step than the π^+ row, consistent with the higher-variance π^{old}-side estimator, and the $16\times$ budget does not smooth them.

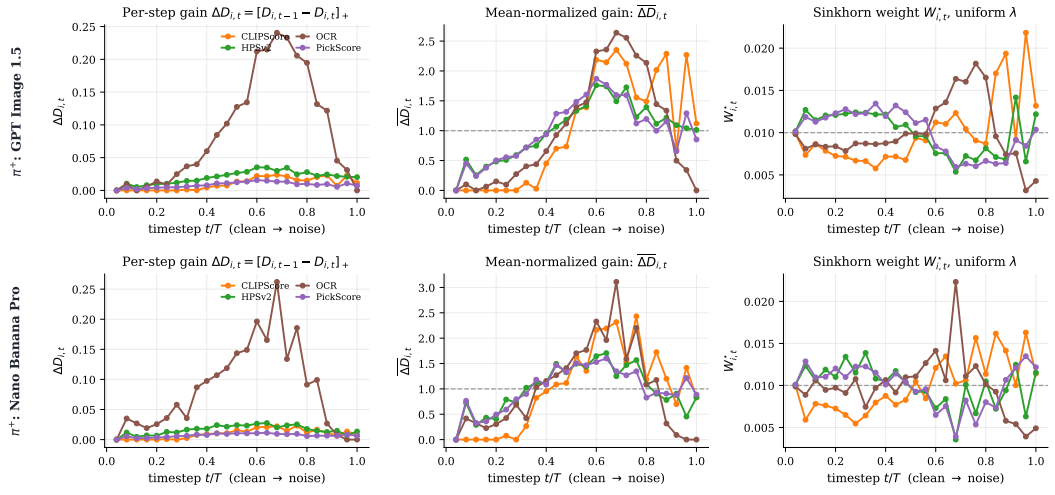


Figure 4: Per-step gain $\Delta D_{i,t}$, mean-normalized gain $\overline{\Delta D}_{i,t}$, and Sinkhorn matrix $W^\star_{i,t}$ at uniform $\lambda_i{=}1/m$, for the training stage-OCR curve under the two π^+ surrogate generators. **Top:** GPT IMAGE 1.5. **Bottom:** NANO BANANA PRO. The two rows produce near-identical reward-specific shapes and peak orderings (quantified in Tab. 5).

B.6 Closing the divergence to π^+ over a training stage

In this section, we measure how much of the divergence the curve is built from survives training against it. We rerun the identical training stage-OCR estimator with π^{old} set to checkpoint-180 of the $\boldsymbol{\lambda}=(1,1,1,1)$ ReCAST stage-OCR run, keeping the GPT Image 1.5 π^+ samples and every other estimator setting fixed, so the policy the gains are measured at is the only thing that changes.

By Eq. (7), the total gain $\sum_t \Delta D_{i,t}$ is the surrogate data-level divergence $D_\alpha\big(\pi^+_{i,0} \,\|\, \pi^{\text{old}}_0\big)$ still separating the policy from the reward-induced positive distribution, so it reads directly as how much of reward i's gap is left to close. Tab. 6 shows that by checkpoint-180 almost none of it is: the total gain falls to $0.14\times$ its base-policy value for ClipScore and to 0.01–$0.11\times$ for the other three rewards, and to $0.03\times$ summed over all four. Fig. 5 shows the same collapse in $D_{i,t}$ itself: on a shared vertical axis, the trained policy's curves sit near zero across the whole trajectory. Because Rényi divergence is nondecreasing in α, the $\alpha{=}2$ quantity we measure upper-bounds $\mathrm{KL}\big(\pi^+_{i,0} \,\|\, \pi^{\text{old}}_0\big)$, so the KL gap to the positive distribution is closed at least as far. Training under the reweighted objective therefore closes almost all of the gap the curve is built from, and it does so for all four rewards simultaneously rather than closing one reward's gap at another's expense.

Table 6: Total discriminability gain $\sum_t \Delta D_{i,t}$ of the training stage-OCR curve with π^{old} at the untrained base policy vs. at the trained checkpoint-180, with the GPT Image 1.5 π^+ samples and all other estimator settings identical. By Eq. (7) each entry is the surrogate data-level divergence remaining between that policy and the reward's positive distribution.

Reward	Base policy	ckpt-180	Remaining
ClipScore	0.252	0.036	$0.14\times$
HPSv2	0.501	0.024	$0.05\times$
OCR	2.278	0.028	$0.01\times$
PickScore	0.209	0.024	$0.11\times$
All four	3.241	0.113	$0.03\times$

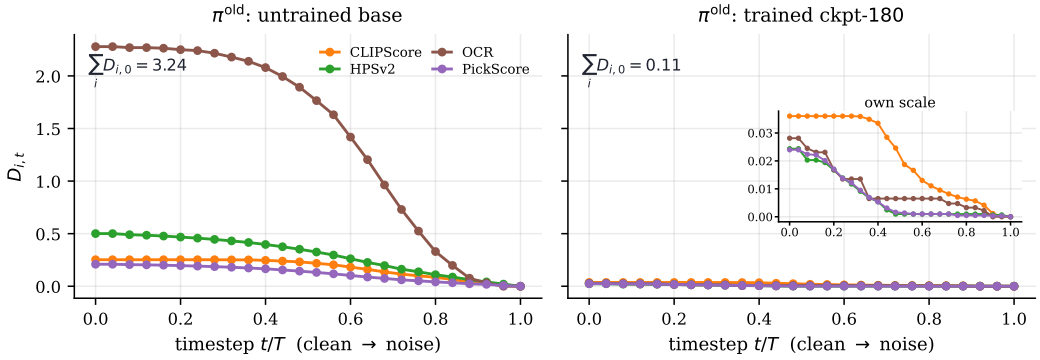


Figure 5: Cumulative discriminability $D_{i,t}$ of the training stage-OCR curve at the two policies, with π^+ set to GPT Image 1.5 and all other estimator settings identical. **Left:** π^{old} at the untrained base. **Right:** π^{old} at checkpoint-180 of the $\boldsymbol{\lambda}=(1,1,1,1)$ ReCAST stage-OCR run, drawn on the same vertical axis; the inset repeats it at its own scale. Each curve telescopes the per-step gains up from $D_{i,T} = 0$ (Eq. (7)), so its clean endpoint $D_{i,0}$ is the full (surrogate) data-level divergence between that policy and reward i's positive distribution. Training leaves $D_{i,t}$ smaller at every timestep, for every reward.

C Sinkhorn projection: solver and alternatives

C.1 The log-domain iteration

Alg. 3 is the solver of Eq. (16). It normalizes the gains per reward, exponentiates them into the kernel, and alternates the two marginal updates of Eq. (19) in the log domain, which keeps the scaling vectors stable when a kernel entry is tiny. Convergence is linear and in practice takes fewer than 100 iterations at $m \le 4$, $T \le 25$; the cost is negligible next to a single training step, and the matrix is computed once per $\boldsymbol{\lambda}$ before training.

Algorithm 3 Sinkhorn construction of the reward-by-timestep weight matrix $\boldsymbol{W}$

Require: Per-step gains $\{\Delta D_{i,t}\}$ (Alg. 1, Sec. 3.3), inter-reward budget $\boldsymbol{\lambda} \in \Delta^{m-1}$, tolerance ε_{tol}, max iterations L

Ensure: weight matrix $\boldsymbol{W} \in \mathcal{U}(\boldsymbol{\lambda})$, used in the loss as $T \cdot W_{i,t}$

for each reward i **do**
  $\overline{\Delta D}_{i,t} \leftarrow \Delta D_{i,t} \,/\, \big(\frac{1}{T}\sum_s \Delta D_{i,s}\big)$ *// Eq. (13): per-reward mean-normalize*
  $\log K_{i,t} \leftarrow \overline{\Delta D}_{i,t}$ *// Eq. (14); $\log K_{i,t} \leftarrow 0$ if reward i has no gain estimate*
end for
$\log \boldsymbol{a} \leftarrow \mathbf{0}_m, \;\; \log \boldsymbol{b} \leftarrow \mathbf{0}_T$
for $\ell = 1, \ldots, L$ **do**
  $\log a_i \leftarrow \log \lambda_i - \operatorname{logsumexp}_t\big(\log K_{i,t} + \log b_t\big)$ *// row marginals $\to \lambda_i$*
  $\log b_t \leftarrow -\log T - \operatorname{logsumexp}_i\big(\log K_{i,t} + \log a_i\big)$ *// column marginals $\to 1/T$*
  $W_{i,t} \leftarrow \exp\big(\log a_i + \log K_{i,t} + \log b_t\big)$
  break if $\max_i \big|\sum_t W_{i,t} - \lambda_i\big| < \varepsilon_{\text{tol}}$ **and** $\max_t \big|\sum_i W_{i,t} - \frac{1}{T}\big| < \varepsilon_{\text{tol}}$
end for
return $\boldsymbol{W}$

C.2 Boundary cases

Our construction sits between the alternatives one might have used instead, and each of them is instructive.

A flat kernel: the static baseline. If no gain curve has any shape in t, then $\overline{\Delta D}_{i,t} \equiv 1$ for every reward, so $K_{i,t} \equiv e$ is rank one and the entropic optimum over $\mathcal{U}(\boldsymbol{\lambda})$ is $W^{\star}_{i,t} = \lambda_i / T$, which is the standard static convex combination.

No entropic term: hard assignment. Dropping the entropy from Eq. (17) leaves an unregularized transport LP, whose solutions are vertices of $\mathcal{U}(\boldsymbol{\lambda})$: each timestep's budget goes essentially to the single reward with the largest normalized gain there. Committing that hard to argmax gains estimated from finite samples is brittle, so we keep the entropic term, which is what makes the solution the smooth, strictly positive matrix of Eq. (18).

An uncurved reward. Because Eq. (13) normalizes each reward to mean 1, the kernel needs no scale or sharpness parameter, and Sec. 3.5's shapes give smooth, non-degenerate matrices. If a reward has no estimated curve at all, we set $K_{i,t} = 1$ for every t: it still receives its full row budget λ_i, spread uniformly up to the shared b_t.

D Per-$\boldsymbol{\lambda}$ results

Tabs. 1 and 2 pool the five budgets; this appendix resolves both of them per $\boldsymbol{\lambda}$, static vs. ReCAST at matched $\boldsymbol{\lambda}$, every entry the mean over the 3 seed replicates with its standard deviation as a subscript. Each budget also carries a Pareto verdict: Ours $\succ$ static if the ReCAST mean is at least as high as the static mean on *every* reward in that table and strictly higher on at least one, static $\succ$ Ours if static (weakly) dominates ReCAST in the same sense, *mixed* otherwise. Every cell scores the run's final checkpoint.

D.1 Training rewards

Tabs. 7 and 8 give the per-$\boldsymbol{\lambda}$ training rewards behind Tab. 1. In training stage-OCR (Tab. 7) ReCAST Pareto-dominates static at 3 of the 5 budgets, namely $(1,1,1,1)$, $(1,1,1,2)$, and $(1,1,2,1)$, and is dominated at none; the aggregate is higher at 4 of 5. The two mixed budgets fail the test on the same coordinate: $(1,2,1,1)$ raises CLIPSCORE, HPSV2, and PICKSCORE while conceding OCR by 0.039, and $(2,1,1,1)$ raises CLIPSCORE and PICKSCORE, ties HPSV2, and concedes OCR by 0.026. This is the per-budget form of the pattern in Tab. 1, where the target reward carries by far the largest seed variance of the four.

Training stage-GenEval (Tab. 8) is a tie budget by budget as well as on average. ReCAST dominates at no $\boldsymbol{\lambda}$, static dominates at none, and all 5 budgets are mixed; at every one of them the aggregate

shifts by less than the seed standard deviation of either method there. The target GENEVAL reward moves in both directions across budgets, up at 3 of 5 and down at $(1, 1, 1, 2)$ and $(1, 2, 1, 1)$, so no budget in this setting resolves the two methods.

D.2 HELD-OUT JUDGES

Tabs. 9 and 10 resolve Tab. 2 the same way. In training stage-OCR (Tab. 9) ReCAST Pareto-dominates static across all four judges at 3 of the 5 budgets, $(1, 1, 2, 1)$, $(1, 2, 1, 1)$, and $(2, 1, 1, 1)$, and is dominated at none. IMAGEREWARD, AESTHETIC, and HPSV3 improve at every budget without exception; the two mixed budgets, $(1, 1, 1, 1)$ and $(1, 1, 1, 2)$, concede only UNIFIEDREWARD-2.0, by 0.025 and 0.017. The generalization gain of Sec. 4.3 is therefore not carried by a subset of the budgets.

Training stage-GenEval (Tab. 10) is the setting whose training rewards tie, and its judge scores favor ReCAST at a majority of budgets: ReCAST dominates at $(1, 1, 1, 2)$, $(1, 1, 2, 1)$, and $(1, 2, 1, 1)$, static dominates at none, and $(1, 1, 1, 1)$ and $(2, 1, 1, 1)$ are mixed. The seed standard deviations here are up to an order of magnitude wider than in training stage-OCR, wide enough that the individual budgets are not resolved by three replicates even where the means separate; the pooled comparison in Tab. 2, which averages the same runs, still favors ReCAST on all four judges.

Table 7: Training stage-OCR, per-$\boldsymbol{\lambda}$ training rewards on the held-out dataset at each run's final checkpoint, static vs. ReCAST. Every entry is the mean over the 3 seed replicates with its standard deviation as a subscript, Aggregate r is the $\boldsymbol{\lambda}$-weighted sum as in Tab. 1, and bold marks the better method per cell, neither when the two agree at the reported precision. Pareto is the verdict over the four training rewards.

$\boldsymbol{\lambda}$	Method	Aggregate r	CLIPSCORE	HPSV2	PICKSCORE	OCR	Pareto
$(1, 1, 1, 1)$	static	$2.387_{\pm 0.058}$	$0.315_{\pm 0.003}$	$0.288_{\pm 0.007}$	$0.890_{\pm 0.002}$	$0.894_{\pm 0.051}$	Ours $\succ$ static
	Ours	$\mathbf{2.435}_{\pm 0.058}$	$\mathbf{0.321}_{\pm 0.001}$	$\mathbf{0.306}_{\pm 0.006}$	$\mathbf{0.900}_{\pm 0.004}$	$\mathbf{0.908}_{\pm 0.055}$	
$(1, 1, 1, 2)$	static	$3.251_{\pm 0.030}$	$0.308_{\pm 0.004}$	$0.255_{\pm 0.022}$	$0.869_{\pm 0.008}$	$0.910_{\pm 0.001}$	Ours $\succ$ static
	Ours	$\mathbf{3.383}_{\pm 0.031}$	$\mathbf{0.319}_{\pm 0.004}$	$\mathbf{0.290}_{\pm 0.022}$	$\mathbf{0.887}_{\pm 0.011}$	$\mathbf{0.944}_{\pm 0.006}$	
$(1, 1, 2, 1)$	static	$3.259_{\pm 0.039}$	$0.313_{\pm 0.003}$	$0.296_{\pm 0.004}$	$0.898_{\pm 0.002}$	$0.854_{\pm 0.035}$	Ours $\succ$ static
	Ours	$\mathbf{3.338}_{\pm 0.043}$	$\mathbf{0.320}_{\pm 0.004}$	$\mathbf{0.306}_{\pm 0.008}$	$\mathbf{0.903}_{\pm 0.005}$	$\mathbf{0.907}_{\pm 0.056}$	
$(1, 2, 1, 1)$	static	$2.709_{\pm 0.130}$	$0.314_{\pm 0.005}$	$0.299_{\pm 0.028}$	$0.889_{\pm 0.010}$	$\mathbf{0.908}_{\pm 0.060}$	mixed
	Ours	$\mathbf{2.720}_{\pm 0.081}$	$\mathbf{0.318}_{\pm 0.002}$	$\mathbf{0.314}_{\pm 0.008}$	$\mathbf{0.906}_{\pm 0.001}$	$0.869_{\pm 0.065}$	
$(2, 1, 1, 1)$	static	$\mathbf{2.727}_{\pm 0.038}$	$0.317_{\pm 0.001}$	$0.302_{\pm 0.008}$	$0.892_{\pm 0.001}$	$\mathbf{0.900}_{\pm 0.030}$	mixed
	Ours	$2.723_{\pm 0.017}$	$\mathbf{0.323}_{\pm 0.003}$	$0.302_{\pm 0.007}$	$\mathbf{0.901}_{\pm 0.005}$	$0.874_{\pm 0.012}$	

Table 8: Training stage-GenEval, per-$\boldsymbol{\lambda}$ training rewards on the held-out dataset at each run's final checkpoint, static vs. ReCAST. Entries, subscripts, bolding, and the Pareto verdict are as in Tab. 7.

$\boldsymbol{\lambda}$	Method	Aggregate r	CLIPSCORE	HPSV2	PICKSCORE	GENEVAL	Pareto
$(1, 1, 1, 1)$	static	$2.390_{\pm 0.003}$	$0.298_{\pm 0.001}$	$\mathbf{0.316}_{\pm 0.005}$	$\mathbf{0.914}_{\pm 0.002}$	$0.861_{\pm 0.002}$	mixed
	Ours	$\mathbf{2.392}_{\pm 0.022}$	$0.298_{\pm 0.003}$	$0.308_{\pm 0.007}$	$0.909_{\pm 0.006}$	$\mathbf{0.877}_{\pm 0.014}$	
$(1, 1, 1, 2)$	static	$\mathbf{3.300}_{\pm 0.017}$	$0.299_{\pm 0.001}$	$\mathbf{0.309}_{\pm 0.004}$	$\mathbf{0.908}_{\pm 0.008}$	$\mathbf{0.892}_{\pm 0.007}$	mixed
	Ours	$3.290_{\pm 0.045}$	$0.299_{\pm 0.000}$	$0.307_{\pm 0.007}$	$0.906_{\pm 0.005}$	$0.889_{\pm 0.019}$	
$(1, 1, 2, 1)$	static	$3.321_{\pm 0.026}$	$\mathbf{0.298}_{\pm 0.001}$	$0.319_{\pm 0.006}$	$\mathbf{0.914}_{\pm 0.007}$	$0.875_{\pm 0.014}$	mixed
	Ours	$\mathbf{3.322}_{\pm 0.014}$	$0.297_{\pm 0.004}$	$\mathbf{0.322}_{\pm 0.001}$	$0.913_{\pm 0.004}$	$\mathbf{0.877}_{\pm 0.009}$	
$(1, 2, 1, 1)$	static	$2.715_{\pm 0.006}$	$0.299_{\pm 0.002}$	$0.313_{\pm 0.003}$	$\mathbf{0.915}_{\pm 0.001}$	$\mathbf{0.876}_{\pm 0.007}$	mixed
	Ours	$2.715_{\pm 0.021}$	$0.299_{\pm 0.002}$	$\mathbf{0.322}_{\pm 0.003}$	$0.908_{\pm 0.003}$	$0.865_{\pm 0.017}$	
$(2, 1, 1, 1)$	static	$\mathbf{2.677}_{\pm 0.071}$	$\mathbf{0.301}_{\pm 0.003}$	$\mathbf{0.309}_{\pm 0.015}$	$0.904_{\pm 0.008}$	$0.863_{\pm 0.042}$	mixed
	Ours	$2.671_{\pm 0.029}$	$0.300_{\pm 0.002}$	$0.302_{\pm 0.013}$	$\mathbf{0.906}_{\pm 0.004}$	$\mathbf{0.864}_{\pm 0.012}$	

E EXPERIMENT DETAILS

E.1 REWARD MODELS

Tab. 11 documents every reward model used anywhere in the paper: its architecture, checkpoint or backbone, and native output range. Two roles recur: *train*, entering the training objective through the weight matrix $\boldsymbol{W}$; *held-out*, scoring checkpoints it never influenced. Sec. 3.5 also estimates gain

Table 9: Training stage-OCR, per-$\boldsymbol{\lambda}$ held-out judge scores at each run's final checkpoint, static vs. ReCAST. Entries, subscripts, bolding, and the Pareto verdict are as in Tab. 7, now over the four judges; the HPSv3 column is at two decimals because its scale is an order of magnitude larger.

$\boldsymbol{\lambda}$	Method	ImageReward	Aesthetic	HPSv3	UnifiedReward-2.0	Pareto
$(1,1,1,1)$	static	$1.433_{\pm 0.056}$	$5.493_{\pm 0.096}$	$12.86_{\pm 0.33}$	$\mathbf{3.179}_{\pm 0.071}$	mixed
	Ours	$\mathbf{1.476}_{\pm 0.027}$	$\mathbf{5.574}_{\pm 0.063}$	$\mathbf{13.16}_{\pm 0.50}$	$3.154_{\pm 0.030}$	
$(1,1,1,2)$	static	$1.341_{\pm 0.060}$	$5.411_{\pm 0.036}$	$12.06_{\pm 0.71}$	$\mathbf{3.146}_{\pm 0.061}$	mixed
	Ours	$\mathbf{1.420}_{\pm 0.096}$	$\mathbf{5.542}_{\pm 0.027}$	$\mathbf{12.44}_{\pm 0.81}$	$3.129_{\pm 0.055}$	
$(1,1,2,1)$	static	$1.438_{\pm 0.042}$	$5.525_{\pm 0.083}$	$13.00_{\pm 0.18}$	$3.159_{\pm 0.008}$	Ours $\succ$ static
	Ours	$\mathbf{1.499}_{\pm 0.029}$	$\mathbf{5.571}_{\pm 0.045}$	$\mathbf{13.25}_{\pm 0.61}$	$\mathbf{3.190}_{\pm 0.065}$	
$(1,2,1,1)$	static	$1.434_{\pm 0.066}$	$5.481_{\pm 0.093}$	$12.67_{\pm 0.54}$	$3.129_{\pm 0.071}$	Ours $\succ$ static
	Ours	$\mathbf{1.517}_{\pm 0.003}$	$\mathbf{5.622}_{\pm 0.075}$	$\mathbf{13.44}_{\pm 0.39}$	$\mathbf{3.181}_{\pm 0.006}$	
$(2,1,1,1)$	static	$1.461_{\pm 0.032}$	$5.521_{\pm 0.025}$	$13.13_{\pm 0.35}$	$3.158_{\pm 0.028}$	Ours $\succ$ static
	Ours	$\mathbf{1.499}_{\pm 0.025}$	$\mathbf{5.581}_{\pm 0.046}$	$\mathbf{13.41}_{\pm 0.40}$	$\mathbf{3.226}_{\pm 0.070}$	

Table 10: Training stage-GenEval, per-$\boldsymbol{\lambda}$ held-out judge scores at each run's final checkpoint, static vs. ReCAST. Entries, subscripts, bolding, and the Pareto verdict are as in Tab. 9.

$\boldsymbol{\lambda}$	Method	ImageReward	Aesthetic	HPSv3	UnifiedReward-2.0	Pareto
$(1,1,1,1)$	static	$\mathbf{1.372}_{\pm 0.011}$	$\mathbf{6.057}_{\pm 0.020}$	$8.73_{\pm 0.51}$	$\mathbf{2.745}_{\pm 0.025}$	mixed
	Ours	$1.358_{\pm 0.046}$	$6.006_{\pm 0.076}$	$\mathbf{9.30}_{\pm 0.12}$	$2.741_{\pm 0.025}$	
$(1,1,1,2)$	static	$1.268_{\pm 0.073}$	$5.882_{\pm 0.128}$	$8.43_{\pm 1.01}$	$2.705_{\pm 0.065}$	Ours $\succ$ static
	Ours	$\mathbf{1.317}_{\pm 0.106}$	$\mathbf{5.938}_{\pm 0.129}$	$\mathbf{8.61}_{\pm 0.72}$	$\mathbf{2.709}_{\pm 0.042}$	
$(1,1,2,1)$	static	$1.234_{\pm 0.203}$	$5.913_{\pm 0.228}$	$8.19_{\pm 1.58}$	$2.698_{\pm 0.125}$	Ours $\succ$ static
	Ours	$\mathbf{1.386}_{\pm 0.061}$	$\mathbf{6.040}_{\pm 0.090}$	$\mathbf{8.70}_{\pm 0.62}$	$\mathbf{2.736}_{\pm 0.030}$	
$(1,2,1,1)$	static	$0.990_{\pm 0.562}$	$5.699_{\pm 0.380}$	$7.23_{\pm 2.74}$	$2.588_{\pm 0.250}$	Ours $\succ$ static
	Ours	$\mathbf{1.369}_{\pm 0.033}$	$\mathbf{6.054}_{\pm 0.037}$	$\mathbf{8.46}_{\pm 0.42}$	$\mathbf{2.706}_{\pm 0.025}$	
$(2,1,1,1)$	static	$\mathbf{1.127}_{\pm 0.190}$	$5.768_{\pm 0.223}$	$7.49_{\pm 1.01}$	$2.618_{\pm 0.096}$	mixed
	Ours	$1.014_{\pm 0.540}$	$\mathbf{5.800}_{\pm 0.337}$	$\mathbf{7.81}_{\pm 2.33}$	$\mathbf{2.633}_{\pm 0.182}$	

curves for two of the held-out judges (Aesthetic, ImageReward), to show what the estimator produces on rewards of a different type; they enter no training kernel and no training loss.

The training-stage kernels are built from OCR's and GenEval's own gain curves, estimated exactly as in Sec. 3.5 but on their own prompt sets and their own $T{=}25$ schedule. Fig. 1 also overlays both, linearly resampled $T{=}25$ onto the $T{=}10$ grid of the other five, purely for visual comparison.

Positivity of the training rewards. The analysis of Sec. 3.2 assumes $r_i \geq 0$, so we state how each training reward is computed. Write $\cos(\boldsymbol{u}, \boldsymbol{v})$ for the cosine similarity of an image embedding $\boldsymbol{u}$ and a text embedding $\boldsymbol{v}$. ClipScore is cos between the CLIP ViT-L/14 image and text embeddings. HPSv2 is cos between the ℓ_2-normalized image and text embeddings of the preference-fine-tuned ViT-H/14, with no logit scale applied. PickScore is the preference head's cos rescaled by the model's own learned logit scale and a fixed constant, $(e^s/26)\cos$ with $e^s \approx 98.9$, so $\approx 3.80\cos$. OCR is $1 - \min\{\mathrm{Lev}(\hat{s}, s), |s|\}/|s|$, where s is the target string quoted in the prompt, $\hat{s}$ is the recognized text after lowercasing and removing whitespace, Lev is the Levenshtein distance, and the distance is taken as 0 whenever s occurs as a substring of $\hat{s}$. GenEval is the fraction of the prompt's object, count, color, and position clauses that the detector confirms.

OCR and GenEval are therefore non-negative by construction, one a distance ratio capped at 1 and the other a fraction of satisfied clauses. The remaining three are cosine similarities up to a positive scale, so they are sign-indefinite in principle, with ranges $[-1, 1]$, $[-1, 1]$, and $\approx [-3.80, 3.80]$. We clamp the reward at 0. This clamping has no effect in estimation, because on our prompt sets they are positive throughout and with a wide margin: over every trained checkpoint we evaluate, the smallest ClipScore is 0.26, the smallest HPSv2 is 0.23, and the smallest PickScore is 0.77, against a floor of 0. This reflects a property of CLIP-style encoders rather than a coincidence, since image and text embeddings occupy separate cones and their cosines concentrate in a narrow positive band. We apply a sigmoid to ImageReward, whose raw output is a signed comparison scalar, when estimating its gain curve in Sec. 3.5.

Table 11: Reward models used in the paper. "Range" is the reward's output scale; several are unbounded in principle but concentrate in the interval shown in practice.

Reward	Architecture / checkpoint	Range	Role
CLIPSCORE (Hessel et al., 2021)	CLIP ViT-L/14 (`openai/clip-vit-large-patch14`) image–text cosine similarity	$\approx [0, 1]$	train
HPSV2 (Wu et al., 2023)	open_clip ViT-H/14 backbone fine-tuned on human preference pairs (`HPS_v2.1_compressed.pt` checkpoint)	$\approx [0, 1]$	train
PICKSCORE (Kirstain et al., 2023)	`yuvalkirstain/PickScore_v1`, a CLIP ViT-H/14 fine-tuned end-to-end on preference pairs, with the `laion/CLIP-ViT-H-14-laion2B-s32B-b79K` processor	$\approx [0, 3.8]$	train
OCR	PaddleOCR (English, angle classification off) text extraction	$[0, 1]$	train (training stage-OCR)
GENEVAL (Ghosh et al., 2023)	Mask2Former (Swin-S, COCO) detector + CLIP zero-shot color/counting classifiers	$[0, 1]$	train (training stage-GenEval)
AESTHETIC (Schuhmann, 2022)	Linear predictor over frozen CLIP ViT-L/14 embeddings, trained on SAC, LAION-Logos, and AVA aesthetic ratings (`sac+logos+ava1-l14-linearMSE` checkpoint)	$\approx [1, 10]$	held-out
IMAGEREWARD (Xu et al., 2023)	Official `ImageReward-v1.0` checkpoint	unbounded, z-scored	held-out
HPSV3 (Ma et al., 2025)	`HPSv3RewardInferencer`, a $\sim$7B-parameter vision-language reward model	unbounded, $\approx [-10, 16]$	held-out
UNIFIEDREWARD-2.0 (Wang et al., 2025)	`CodeGoat24/UnifiedReward-2.0-qwen3vl-8b`, a Qwen3-VL-8B pointwise judge served through vLLM (greedy decoding, temperature 0)	$[1, 5]$	held-out

E.2 TRAINING HYPERPARAMETERS AND COMPUTE

Architecture and optimization. All runs fine-tune LoRA adapters (rank 32, scale 64, Gaussian initialization) inserted into the eight attention-projection modules of the SD3.5-Medium joint-attention transformer block; the transformer backbone itself is frozen. The optimizer is AdamW with learning rate 3×10^{-4}, $(\beta_1, \beta_2) = (0.9, 0.999)$, $\epsilon = 10^{-8}$, weight decay 10^{-4}, and a decayed learning-rate schedule (`decay_type=1`) shared by every stage. The DiffusionNFT coefficient is $\beta = 0.1$ throughout, sampling uses $T = 25$ steps, and the rollout batch is 24 images/prompt on each of 8 GPUs with 6 gradient-accumulation steps, for an effective batch of $8 \times 24 \times 6 = 1152$. Evaluation images are drawn at seed 42+index, identical across the two methods of a pair. Sinkhorn kernels are built once per $\boldsymbol{\lambda}$ before training starts, at $\alpha = 2$ for every training reward.

Compute. All training ran on single-node 8×NVIDIA H200 allocations. A warmup-stage run costs ≈ 68 GPU-hours, a training stage-OCR run ≈ 37, and a training stage-GenEval run ≈ 59. The static and ReCAST methods of a pair cost the same: the projection is negligible next to sampling and reward scoring.

Gain curve construction cost. Each curve uses N=8 π^+ samples per prompt, and K=16 π^{old} rollouts from every intermediate $\boldsymbol{x}_t$; every reward of a stage is scored on the same rollouts, so the cost is per curve rather than per reward, and App. F reports what reproducing one curve from scratch costs. Ignoring the GPT IMAGE 1.5 API time, the $P{\cdot}N{\cdot}T{\cdot}K$ rollouts of one T=10 curve take ≈ 15 GPU-hours, under half the $\approx$ 37 of a single training stage-OCR run, and a training-stage curve at T=25 takes about 6× those 15 GPU-hours. The $P{\cdot}N$ queries to GPT IMAGE 1.5 cost around $35 in API fees. Crucially, this is a one-time cost per stage: each curve is estimated once at the untrained policy and then reused, unmodified, so its amortized cost per training run is well below the raw totals.

E.3 MMRB2-DERIVED JUDGE: PROTOCOL, POSITION DEBIASING, AND PROMPT

Protocol. The pairwise judgments of Sec. 4.4 adapt the image-generation judge prompt and evaluation protocol from Meta's MMRB2 (Hu et al., 2025) suite, and serve the resulting judge with `gemini-3.5-flash` at temperature 0. For each of the five matched $\boldsymbol{\lambda}$ budgets, we pair the static method and the ReCAST method image-by-image on the fixed 1,000-prompt subset of the held-out OCR dataset of Tab. 2. Each API call sends the rubric prompt below, the original text prompt, and the two images labeled `[RESPONSE A:]` and `[RESPONSE B:]`. The judge returns a structured JSON verdict: step-by-step reasoning over seven criteria (faithfulness to prompt, text rendering, input faithfulness, image consistency, text–image alignment, text quality, overall quality), a scalar score $s \in \{1, \ldots, 6\}$ where $s \geq 4$ favors the image shown as A, a discrete `better_response` $\in$ {A, B}, and a self-reported confidence. We take the winner from `better_response`; calls that still fail after 8 retries drop the pair.

Position debiasing. LLM judges exhibit position bias: the response presented first is systematically favored. Following MMRB2's forward+reverse protocol, every pair is therefore judged *twice*: once with the static image shown as A and ReCAST as B (forward), and once with the images swapped (reverse). Each ordering credits one win to the method the judge picks, after mapping the shown-side label back to the method's identity. The win rate over n prompts is the fraction of the $2n$ orderings won,

$$\text{win(Ours)} \;=\; \frac{1}{2n} \sum_{j=1}^{n} \#\{\text{orderings of pair } j \text{ won by Ours}\}.$$

Faithfulness and aesthetic sub-scores. The judge is not asked for numeric per-criterion scores; the *Faithfulness* and *Aesthetics* rows of Tab. 3 are extracted from its per-criterion reasoning fields `faithfulness_to_prompt` and `overall_quality`, respectively. For each ordering, the criterion's reasoning text is mapped to a winner in {A, B, undecided} by matching declarative phrasings like "Response A is better…" and "the edge goes to B…", with a text-only LLM tie-break when the patterns are ambiguous. The sub-score is the fraction of *decided* orderings won by ReCAST; or-

derings whose criterion text is undecided or marked not-applicable are dropped from that criterion's denominator.

Judge prompt. The rubric prompt, verbatim from MMRB2's image-generation judge:

```
You are an expert in multimodal quality analysis and generative AI evaluation. Your
role is to act as an objective judge for comparing two AI-generated responses to the
same prompt. You will evaluate which response is better based on a comprehensive
rubric.

**Important Guidelines:**
- Be completely impartial and avoid any position biases
- Ensure that the order in which the responses were presented does not influence
  your decision
- Do not allow the length of the responses to influence your evaluation
- Do not favor certain model names or types
- Be as objective as possible in your assessment
- Consider factors such as helpfulness, relevance, accuracy, depth, creativity, and
  level of detail

**Understanding the Content Structure:**
- **[ORIGINAL PROMPT TO MODEL:]**: This is the instruction given to both AI models
- **[INPUT IMAGE FROM PROMPT:]**: This is the source image provided to both models
  (if any)
- **[RESPONSE A:]**: The first model's generated response (text and/or images)
- **[RESPONSE B:]**: The second model's generated response (text and/or images)

Your evaluation must be based on a fine-grained rubric that covers the following
criteria. For each criterion, you must provide detailed step-by-step reasoning
comparing both responses. You will use a 1-6 scoring scale.

**Evaluation Criteria:**
1. **faithfulness_to_prompt:** Which response better adheres to the composition,
   objects, attributes, and spatial relationships described in the text prompt?

2. **text_rendering:** If either response contains rendered text, which one has
   better text quality (spelling, legibility, integration)? If no text is rendered,
   state "Not Applicable."

3. **input_faithfulness:** If an input image is provided, which response better
   respects and incorporates the key elements and style of that source image? If no
   input image is provided, state "Not Applicable."

4. **image_consistency:** If multiple images are generated, which response has
   better visual consistency between images (character appearance, scene details)?
   If no multiple images are provided, state "Not Applicable."

5. **text_image_alignment:** Which response has better alignment between text
   descriptions and visual content?

6. **text_quality:** If text was generated, which response has better linguistic
   quality (correctness, coherence, grammar, tone)?

7. **overall_quality:** Which response has better general technical and aesthetic
   quality, realism, coherence, and fewer visual artifacts or distortions?

**Scoring Rubric:**
- Score 6 (A is significantly better): Response A is significantly superior across
  most criteria
- Score 5 (A is marginally better): Response A is noticeably better across several
  criteria
- Score 4 (Unsure or A is negligibly better): Response A is slightly better or
  roughly equivalent
- Score 3 (Unsure or B is negligibly better): Response B is slightly better or
  roughly equivalent
- Score 2 (B is marginally better): Response B is noticeably better across several
  criteria
- Score 1 (B is significantly better): Response B is significantly superior across
  most criteria

**Confidence Assessment:**
After your evaluation, assess your confidence in this judgment on a scale of 0.0
to 1.0:

**CRITICAL**: Be EXTREMELY conservative with confidence scores. Most comparisons
should be in the 0.2-0.5 range.

- **Very High Confidence (0.8-1.0)**: ONLY for absolutely obvious cases where one
  response is dramatically better across ALL criteria with zero ambiguity. Use this
```

```
  extremely rarely (less than 10% of cases).
- **High Confidence (0.6-0.7)**: Clear differences but some uncertainty remains.
  Use sparingly (less than 20% of cases).
- **Medium Confidence (0.4-0.5)**: Noticeable differences but significant
  uncertainty. This should be your DEFAULT range.
- **Low Confidence (0.2-0.3)**: Very close comparison, difficult to distinguish.
  Responses are roughly equivalent or have conflicting strengths.
- **Very Low Confidence (0.0-0.1)**: Essentially indistinguishable responses or
  major conflicting strengths.

**IMPORTANT GUIDELINES**:
- DEFAULT to 0.3-0.5 range for most comparisons
- Only use 0.6+ when you are absolutely certain
- Consider: Could reasonable people disagree on this comparison?
- Consider: Are there any strengths in the "worse" response?
- Consider: How obvious would this be to a human evaluator?
- Remember: Quality assessment is inherently subjective

After your reasoning, you will provide a final numerical score, indicate which
response is better, and assess your confidence. You must always output your
response in the following structured JSON format:

{
    "reasoning": {
        "faithfulness_to_prompt": "YOUR REASONING HERE",
        "text_rendering": "YOUR REASONING HERE",
        "input_faithfulness": "YOUR REASONING HERE",
        "image_consistency": "YOUR REASONING HERE",
        "text_image_alignment": "YOUR REASONING HERE",
        "text_quality": "YOUR REASONING HERE",
        "overall_quality": "YOUR REASONING HERE",
        "comparison_summary": "YOUR OVERALL COMPARISON SUMMARY HERE"
    },
    "score": <int 1-6>,
    "better_response": "A" or "B",
    "confidence": <float 0.0-1.0>,
    "confidence_rationale": "YOUR CONFIDENCE ASSESSMENT REASONING HERE"
}
```

F LIMITATIONS

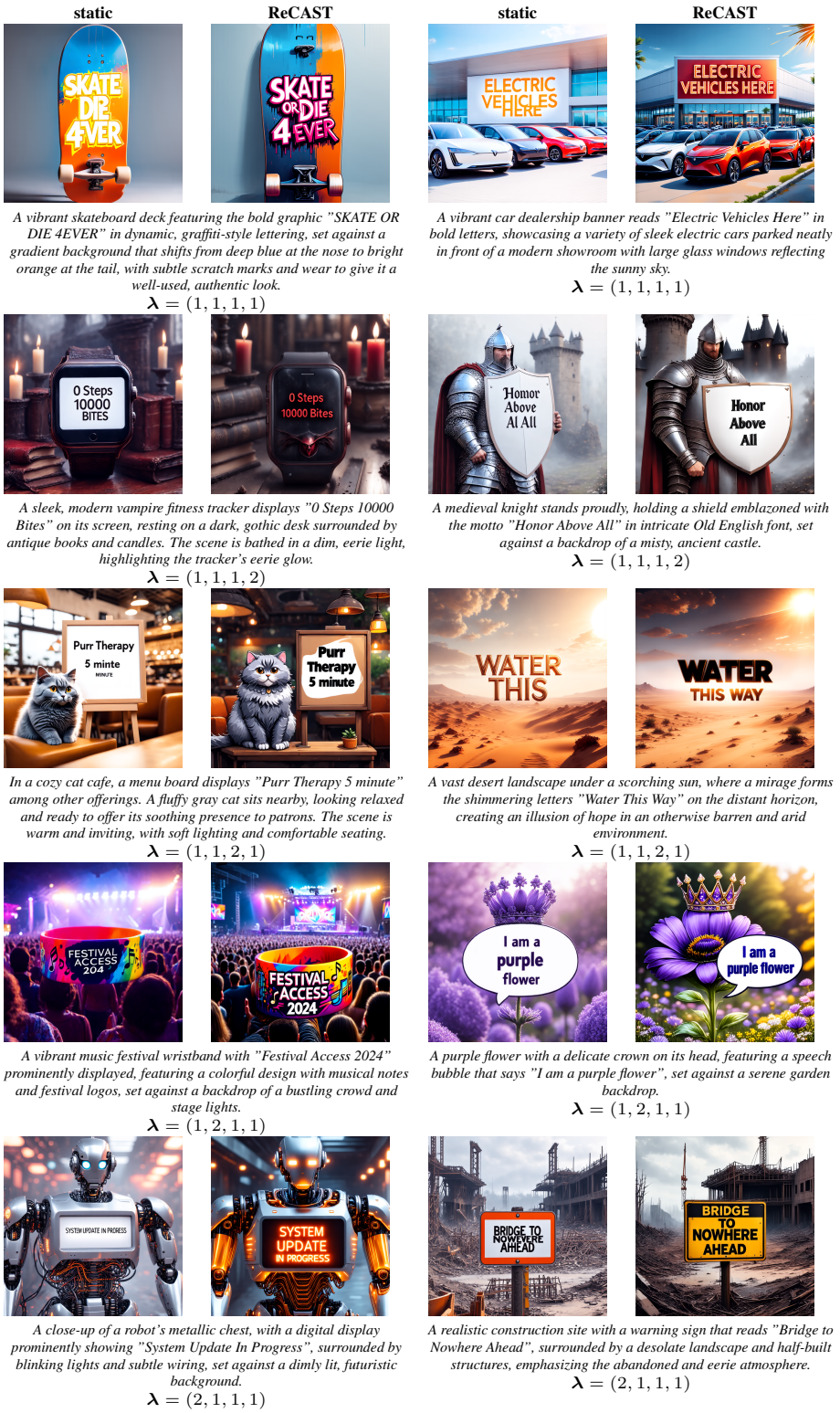
Approximate positive-policy sampling. In practice, we use GPT IMAGE 1.5, an external expert generator, as a surrogate for π^+ for every reward. Its distribution need not equal any reward-specific π_i^+; without an importance correction, the resulting gain curves are therefore biased estimates of the theoretical $\Delta D_{i,t}$. The empirical curves and ablations show that this proxy is useful, and swapping in an independent surrogate generator leaves the curve essentially unchanged (App. B.5), but neither removes this bias. Future work could construct reward-specific positive policies, estimate the required density correction, or update the curves on-policy as training progresses; App. B.6 re-estimates the curve at a trained checkpoint.

Reproducibility and API cost. Depending on GPT IMAGE 1.5 also means depending on a proprietary, versioned, and priced API. Even though it is a one-time, per-stage expense amortized over many downstream training runs, an independent group reproducing a single curve from scratch pays around $35 in API costs. Releasing the estimated gain-curve weight files themselves alongside the code would let downstream use of ReCAST skip re-querying GPT IMAGE 1.5 entirely.

G QUALITATIVE EXAMPLES

The two galleries below (Figs. 6 and 7) show paired generations from the static baseline and ReCAST for the OCR and GENEVAL settings. We state explicitly how these examples were selected so that they are not mistaken for random or typical samples. Every one of the 20 pairs shown (2 per $\boldsymbol{\lambda}$ budget, with 5 budgets per setting) is a decisive, judge-verified win for ReCAST, drawn from the pool of "high-confidence" pairs identified by MMRB2-style pairwise-judge runs following the protocol in Sec. 4.4 and App. E.3, applied separately to each setting's own held-out prompts. For the GENEVAL setting, this selection uses an analogous MMRB2-style run whose aggregate results are not reported in the main text, since Sec. 4.4 reports only the OCR setting. We call a win *decisive* when ReCAST wins under both presentation orderings with extreme scores and neither the faithful-

ness nor the aesthetic sub-criterion contradicts the overall verdict. Within each budget's qualifying pool, the displayed pair was then selected by a human reviewer rather than sampled uniformly at random. These galleries should therefore be interpreted as illustrating what ReCAST's most decisive wins look like, rather than its average-case behavior. Representative aggregate results are reported in Tab. 3 for the OCR setting and in Tabs. 1 and 2, which pool all three seeds, for the GENEVAL setting.



A vibrant skateboard deck featuring the bold graphic "SKATE OR DIE 4EVER" in dynamic, graffiti-style lettering, set against a gradient background that shifts from deep blue at the nose to bright orange at the tail, with subtle scratch marks and wear to give it a well-used, authentic look.
$\boldsymbol{\lambda} = (1, 1, 1, 1)$

A vibrant car dealership banner reads "Electric Vehicles Here" in bold letters, showcasing a variety of sleek electric cars parked neatly in front of a modern showroom with large glass windows reflecting the sunny sky.
$\boldsymbol{\lambda} = (1, 1, 1, 1)$

A sleek, modern vampire fitness tracker displays "0 Steps 10000 Bites" on its screen, resting on a dark, gothic desk surrounded by antique books and candles. The scene is bathed in a dim, eerie light, highlighting the tracker's eerie glow.
$\boldsymbol{\lambda} = (1, 1, 1, 2)$

A medieval knight stands proudly, holding a shield emblazoned with the motto "Honor Above All" in intricate Old English font, set against a backdrop of a misty, ancient castle.
$\boldsymbol{\lambda} = (1, 1, 1, 2)$

In a cozy cat cafe, a menu board displays "Purr Therapy 5 minute" among other offerings. A fluffy gray cat sits nearby, looking relaxed and ready to offer its soothing presence to patrons. The scene is warm and inviting, with soft lighting and comfortable seating.
$\boldsymbol{\lambda} = (1, 1, 2, 1)$

A vast desert landscape under a scorching sun, where a mirage forms the shimmering letters "Water This Way" on the distant horizon, creating an illusion of hope in an otherwise barren and arid environment.
$\boldsymbol{\lambda} = (1, 1, 2, 1)$

A vibrant music festival wristband with "Festival Access 2024" prominently displayed, featuring a colorful design with musical notes and festival logos, set against a backdrop of a bustling crowd and stage lights.
$\boldsymbol{\lambda} = (1, 2, 1, 1)$

A purple flower with a delicate crown on its head, featuring a speech bubble that says "I am a purple flower", set against a serene garden backdrop.
$\boldsymbol{\lambda} = (1, 2, 1, 1)$

A close-up of a robot's metallic chest, with a digital display prominently showing "System Update In Progress", surrounded by blinking lights and subtle wiring, set against a dimly lit, futuristic background.
$\boldsymbol{\lambda} = (2, 1, 1, 1)$

A realistic construction site with a warning sign that reads "Bridge to Nowhere Ahead", surrounded by a desolate landscape and half-built structures, emphasizing the abandoned and eerie atmosphere.
$\boldsymbol{\lambda} = (2, 1, 1, 1)$

Figure 6: Qualitative examples from the OCR setting (training stage-OCR family, held-out 1000-prompt OCR dataset of Tab. 2): two prompts per $\boldsymbol{\lambda}$ budget, static baseline (left of each pair) vs. ReCAST (right of each pair). All 10 pairs are a curated selection of decisive, judge-verified ReCAST wins, not a random or representative sample; see App. G for the selection methodology and Tab. 3 for this setting's aggregate win rates.

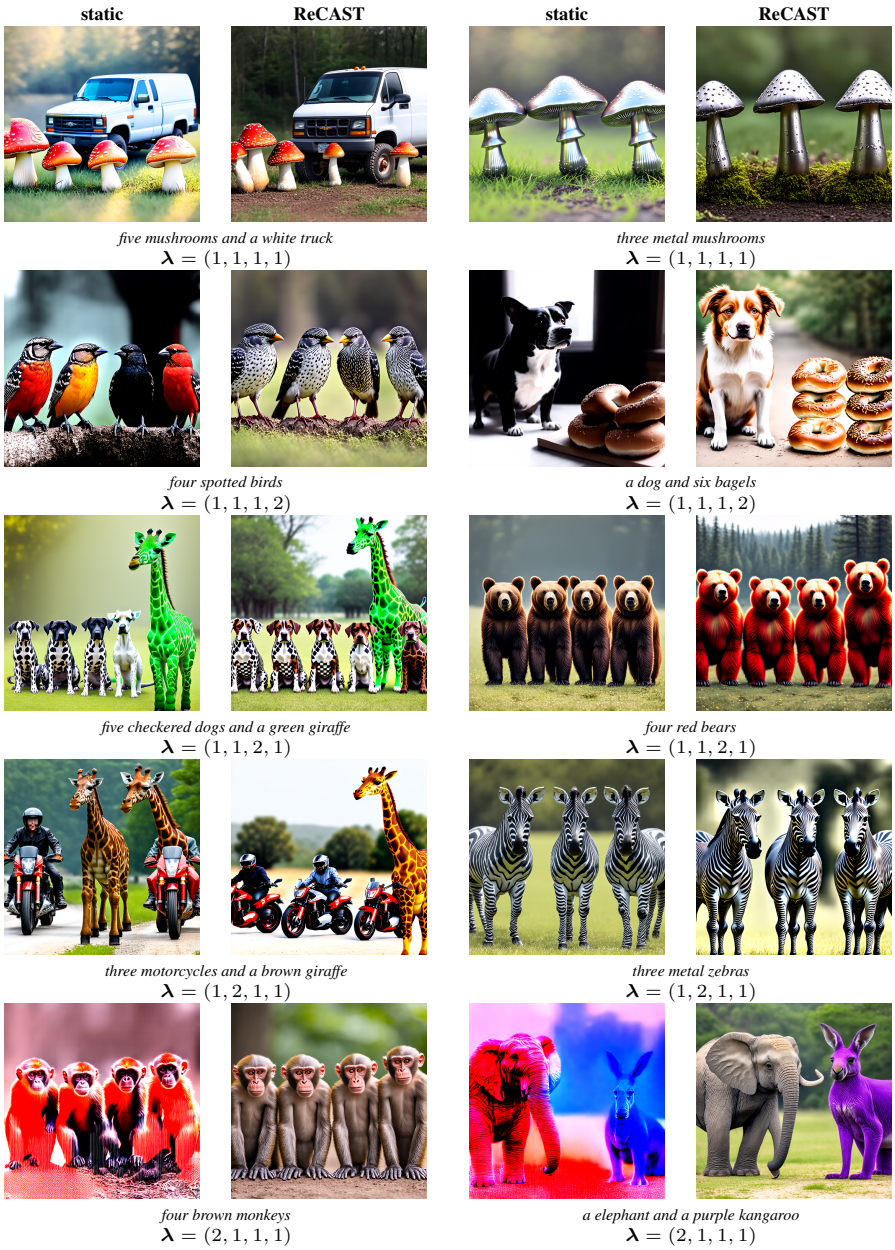


Figure 7: Qualitative examples from the GENEVAL setting (training stage-GenEval family, held-out GENEVAL prompts): two prompts per $\boldsymbol{\lambda}$ budget, static baseline (left of each pair) vs. ReCAST (right of each pair). All 10 pairs are a curated selection of decisive, judge-verified ReCAST wins, not a random or representative sample; see App. G for the selection methodology and Tabs. 1 and 2 for this setting's aggregate results; Sec. 4.4 reports the pairwise judge for the OCR setting only.