DISTRIBUTION-BASED REGULARIZATION FOR REWARD FINE-TUNING DIFFUSION AND FLOW MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Reward fine-tuning is an effective way to adapt pretrained diffusion and flow-matching models to downstream objectives, but aggressive optimization often leads to mode collapse and distributional drift. Existing methods attempt to control this drift with reverse-KL regularization, implemented as an ℓ_2 penalty on the velocity field; however, this functions by indirectly constraining sampling dynamics rather than directly acting on the generated distribution. To address this, we propose DFR (Distributional Functional Rewards), which augments the task reward with a distribution-dependent, per-sample reward that acts explicitly on the generated distribution. Formally, we generalize the KL-regularized objective from expected rewards, which are linear in the model's distribution, to convex functionals of it. The first variation of this objective defines an effective, distribution-dependent reward that directly penalizes deviation from the pretrained distribution. This formulation yields additive per-sample rewards for the Fréchet distance (FD), maximum mean discrepancy (MMD), and DreamSim diversity, recovering several recently proposed rewards as special cases. Across various text-to-image fine-tuning tasks, DFR matches the peak reward of KL-tuned baselines while increasing diversity by up to 29% at equal or lower drift.

1 INTRODUCTION

Diffusion and flow-matching models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021; Lipman et al., 2023; Liu et al., 2023; Albergo et al., 2023; Peluchetti, 2023) have become a dominant paradigm for generative modeling across various data modalities, including images (Rombach et al., 2022; Labs et al., 2025), videos (Wan et al., 2025), and molecular structures (Abramson et al., 2024). These models learn stochastic or deterministic dynamics that transport a simple source distribution to a complex target distribution. However, downstream applications rarely value all samples equally: the goal is often to shift probability mass toward high-reward samples, while preserving the diversity and capabilities of the pretrained model. *Reward fine-tuning* achieves this by adapting the pretrained model toward a downstream reward (Uehara et al., 2024).

For non-differentiable rewards, existing approaches broadly fall into two families: (1) *Trajectory*-based methods, which formulate sampling as a multi-step Markov decision process (MDP), treating each step of the sampling process as an action (Black et al., 2024; Fan et al., 2023); and (2) *ELBO*-based methods, which instead treat the final sample as the action and approximate its intractable log-likelihood with an evidence lower bound (McAllister et al., 2026). Recently, ELBO-based methods have emerged as particularly attractive options for large-scale text-to-image fine-tuning, reporting 5–25$\times$ faster training than trajectory-based methods at comparable reward and image quality (Zheng et al., 2026; Xue et al., 2026; Bergmeister et al., 2026).

Across both families, however, reward maximization alone does not specify which properties of the pretrained distribution ought to be preserved. In particular, optimizing toward increasingly high-reward samples can reduce sample diversity and move the generated distribution away from the pretrained model. Existing methods often address this tension through KL regularization, typically implemented as an ℓ_2 penalty between the fine-tuned and pretrained velocity fields (Fan et al., 2023; Liu et al., 2025b). However, this regularization acts on the model's *sampling dynamics*, whereas the properties of ultimate interest—diversity and retention of the pretrained distribution—are properties of its *output distribution*. This raises the natural question:

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

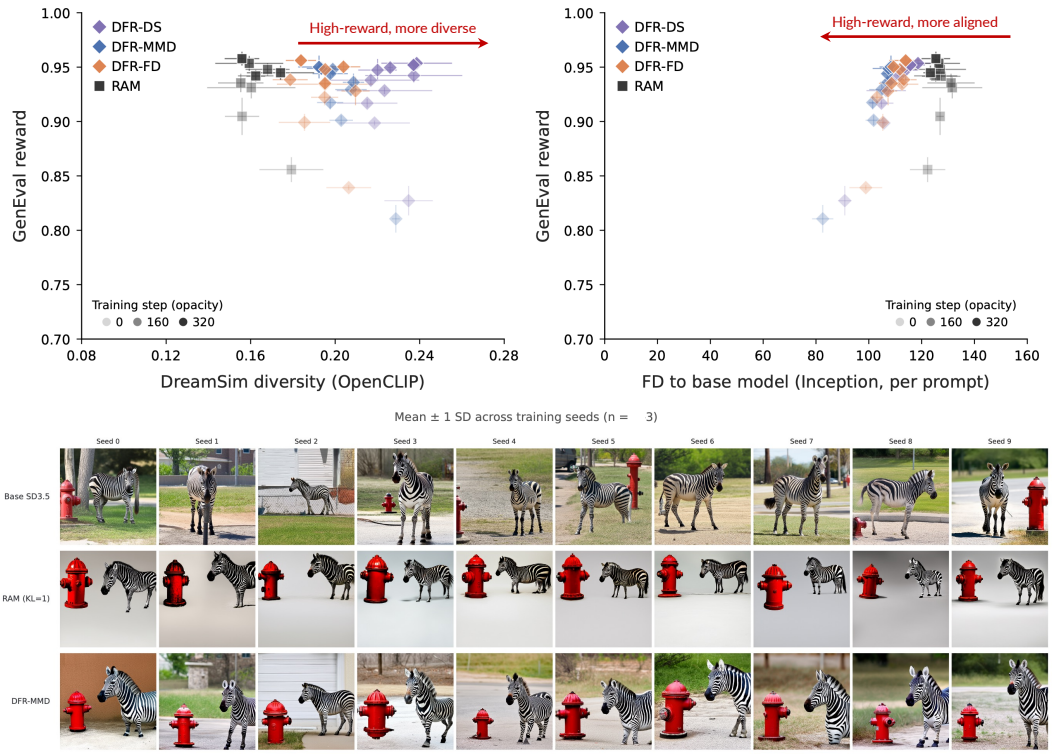


Figure 1: **Top**: RAM achieves high rewards at the cost of sample diversity and distributional fidelity, whereas DFR matches—and often surpasses—its rewards while preserving diversity and FID relative to the pretrained distribution. **Bottom**: DFR more closely mirrors the pretrained SD3.5 distribution, generating a wider diversity of images relative to RAM across multiple seeds for the same prompt: *"A zebra to the right of a fire hydrant."*

Can we regularize the generated distribution directly during reward fine-tuning?

We answer this question affirmatively with Distributional Functional Rewards (DFR), a general framework for distribution-regularized reward fine-tuning. DFR unifies trajectory- and ELBO-based methods by augmenting reward optimization with objectives defined directly on the generated distribution. This allows models to improve reward while better preserving the pretrained distribution. Our main contributions are:

- **A unified framework for distribution-regularized reward fine-tuning.** DFR generalizes KL-regularized reward optimization to convex functionals of the clean-sample distribution, including maximum mean discrepancy (MMD), Fréchet distance (FD), and diversity objectives. Their first variations induce distribution-dependent per-sample reward corrections, enabling direct control of the generated distribution in both trajectory- and ELBO-based methods.
- **Improved distribution preservation at matched reward.** On GenEval and OCR with SD3.5-M, DFR reduces drift from the pretrained model by up to 12% and increases diversity by up to 29% relative to KL-regularized baselines at matched reward, while training 36–41$\times$ faster than Flow-GRPO.
- **Compatibility with both feature- and latent-space objectives.** We instantiate DFR with feature-space objectives such as FD and DreamSim diversity, as well as latent-space MMD, demonstrating strong performance even without the use of pretrained feature encoders.
- **Minimal overhead over KL-regularized fine-tuning.** Compared with KL-tuned RAM, DFR adds little training cost: DFR-MMD reaches GenEval 0.95 with only 7% more training time than RAM ($\beta = 3$) while achieving 9% higher diversity, and DFR-DS reaches OCR 0.96 with only 1% more time than RAM ($\beta = 5$) while achieving 17% higher diversity.

2 BACKGROUND

2.1 FLOW-BASED MODELS

Flow-based models—including diffusion and flow-matching frameworks (Lipman et al., 2023; Song et al., 2021)—learn a time-dependent velocity field $v_\theta(\boldsymbol{x}_t, t)$ that transports samples between a source distribution, typically Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$, and the data distribution $\boldsymbol{x}_0 \sim p_{\text{data}}$. For the commonly used linear path $\boldsymbol{x}_t = (1-t)\boldsymbol{x}_0 + t\boldsymbol{\epsilon}$, the corresponding target velocity is simply $\dot{\boldsymbol{x}}_t = \boldsymbol{\epsilon} - \boldsymbol{x}_0$. The velocity field is therefore learned by weighted regression toward this target:

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{\epsilon}}\left[w(t)\,\|v_\theta(\boldsymbol{x}_t, t) - (\boldsymbol{\epsilon} - \boldsymbol{x}_0)\|_2^2\right], \tag{1}$$

where $w(t) > 0$ is a timestep-dependent weighting function. At inference, samples are generated by drawing $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$ and integrating the learned dynamics backward from $t = 1$ to $t = 0$. We denote the frozen pretrained velocity field by v_{ref} and the endpoint distribution induced by v_θ under the chosen sampler by $p_\theta(\boldsymbol{x}_0)$, omitting conditioning for brevity. This integration takes the form of an SDE for diffusion models and an ODE for flow-matching models; we refer the reader to Tong et al. (2023); Liu et al. (2025b) for a more detailed equivalence between the two formulations.

2.2 REWARD FINE-TUNING VIA POLICY GRADIENTS

Reward fine-tuning adapts a pretrained generative model to a downstream reward while regularizing deviations from the pretrained distribution.[1] This is most often formulated as KL-regularized reward maximization:

$$\max_\theta \; \mathbb{E}_{\boldsymbol{x}_0\sim p_\theta}[r(\boldsymbol{x}_0)] - \beta\mathcal{D}_{\text{KL}}\left(p_\theta(\boldsymbol{x}_0)\,\|\,p_{\text{ref}}(\boldsymbol{x}_0)\right), \tag{2}$$

where $\beta > 0$ controls the reverse-KL regularization strength, and p_θ and p_{ref} are the endpoint distributions of the fine-tuned and pretrained models, respectively. For non-differentiable rewards, the gradient of the expected rewards can be evaluated using the REINFORCE identity (Williams, 1992):

$$\nabla_\theta \mathbb{E}_{\boldsymbol{x}_0\sim p_\theta}[r(\boldsymbol{x}_0)] = \mathbb{E}_{\boldsymbol{x}_0\sim p_\theta}\left[(r(\boldsymbol{x}_0) - b)\nabla_\theta \log p_\theta(\boldsymbol{x}_0)\right], \tag{3}$$

where b is a sample-independent baseline used to reduce variance, and the difference $r(\boldsymbol{x}_0) - b$ defines the advantage. In practice, advantages are often normalized within groups of samples from the same prompt (Shao et al., 2024), while clipped likelihood ratios can further stabilize training optimization (Schulman et al., 2017).

For flow-based models, however, the marginal log-likelihood $\log p_\theta(\boldsymbol{x}_0)$ is intractable. Trajectory-based methods replace it with the per-step transition likelihoods of a stochastic sampler (Section 5). ELBO-based methods instead replace it with $-\mathcal{L}_{\text{FM}}(\theta; \boldsymbol{x}_0)$, the loss of Eq. (1) evaluated on a single generated sample. For suitable $w(t)$, this is an evidence lower bound up to an additive constant (McAllister et al., 2026; Choi et al., 2026). Substituting it into Eq. (3) gives online reward-weighted regression (RWR): the model is trained with the flow-matching loss on its own samples, each weighted by its advantage (Fan et al., 2025; Xue et al., 2026). We build on RAM (Bergmeister et al., 2026), which takes this form. See Appendix B for a detailed derivation of these equivalences.

Regularization. While trajectory- and ELBO-based methods differ in how they estimate the reward gradient, both typically approximate the reverse-KL term in Eq. (2) using a squared velocity anchor:

$$\mathcal{L}_{\text{anchor}}(\theta) = \mathbb{E}\left[\|v_\theta(\boldsymbol{x}_t, t) - v_{\text{ref}}(\boldsymbol{x}_t, t)\|_2^2\right]. \tag{4}$$

For Gaussian transition kernels, a time-weighted form of Eq. (4) equals the KL divergence between the path distributions of the fine-tuned and pretrained samplers, which upper-bounds the reverse-KL divergence of Eq. (2) (Fan et al., 2023, Equation 7).

2.3 DISTRIBUTIONAL MEASURES

Fréchet distance (FD). For a feature encoder ϕ, let μ_p and Σ_p denote the mean and covariance of $\phi(\boldsymbol{x})$ under $\boldsymbol{x} \sim p$. The Fréchet distance (FD) compares the first two feature moments of p and q:

$$\text{FD}(p, q) \triangleq \|\mu_p - \mu_q\|_2^2 + \text{tr}\Big(\Sigma_p + \Sigma_q - 2\big(\Sigma_q^{1/2}\Sigma_p\Sigma_q^{1/2}\big)^{1/2}\Big). \tag{5}$$

[1] See Uehara et al. (2025) for a review of inference-time reward alignment.

Maximum mean discrepancy (MMD) is a classical kernel two-sample discrepancy (Gretton et al., 2006). For a chosen kernel $k(\cdot,\cdot)$,

$$\mathrm{MMD}_k^2(p,q) = \mathbb{E}_{\boldsymbol{x},\boldsymbol{x}'\sim p}[k(\boldsymbol{x},\boldsymbol{x}')] + \mathbb{E}_{\boldsymbol{y},\boldsymbol{y}'\sim q}[k(\boldsymbol{y},\boldsymbol{y}')] - 2\mathbb{E}_{\substack{\boldsymbol{x}\sim p\\ \boldsymbol{y}\sim q}}[k(\boldsymbol{x},\boldsymbol{y})], \tag{6}$$

where a characteristic kernel ensures $\mathrm{MMD}_k(p,q) = 0$ if and only if $p = q$. Note, MMD requires no feature encoder and can be computed directly in latent space.

DreamSim diversity. DreamSim (Fu et al., 2023) is a perceptual distance trained to align with human judgments of visual similarity. Given the normalized DreamSim feature $z(\boldsymbol{x}) = \phi_{\mathrm{DS}}(\boldsymbol{x})/\|\phi_{\mathrm{DS}}(\boldsymbol{x})\|_2$, DreamSim diversity (Domingo-Enrich et al., 2025) is the mean pairwise cosine distance between independent samples, which equals the total variance of the normalized features:

$$\mathrm{Div}_{\mathrm{DS}}(p) = \mathbb{E}_{\boldsymbol{x},\boldsymbol{x}'\sim p}\big[1 - z(\boldsymbol{x})^\top z(\boldsymbol{x}')\big] = 1 - \big\|\mathbb{E}_{\boldsymbol{x}\sim p}[z(\boldsymbol{x})]\big\|_2^2 = \mathrm{tr}\ \mathrm{Cov}_{\boldsymbol{x}\sim p}[z(\boldsymbol{x})]. \tag{7}$$

3 DFR: Distributional Regularization for Reward Fine-Tuning

The velocity anchor in Eq. (4) only indirectly controls the reverse-KL divergence in Eq. (2), as it penalizes changes in the sampling dynamics even when the endpoint distribution p_θ is unchanged. In ELBO-based methods, heuristic time weighting at forward-noised states further weakens this correspondence. We nevertheless retain the velocity anchor for training stability and to limit drift, yet we complement it with a term that acts directly on the generated distribution. Importantly, the two objectives are compatible: the optimal fine-tuned process preserves the pretrained model's conditional noising process (demonstrated in Appendix C.2).

We generalize the KL-regularized objective of Eq. (2) from expected rewards to convex functionals of the generated distribution (Section 3.1). The first variation of such a functional acts as a per-sample reward, which any policy gradient method can optimize without further changes (Section 3.2), and instantiate the framework with MMD, FD, and DreamSim diversity (Section 3.3).

3.1 Distributional regularization through convex functionals

Let $\mathcal{F} : \mathcal{P}(\mathcal{X}) \to \mathbb{R}$ be a convex functional of the generated distribution. Consider the entropy-regularized optimization problem

$$p^\star = \arg\min_{p\in\mathcal{P}(\mathcal{X})} \left(\mathcal{F}(p) + \beta\, \mathcal{D}_{\mathrm{KL}}(p\|p_{\mathrm{ref}})\right), \qquad \beta > 0. \tag{8}$$

For a signed measure η on $\mathcal{X}$ with $\eta(\mathcal{X}) = 0$, the first-order change of $\mathcal{F}$ along η defines the *functional derivative* $\frac{\delta\mathcal{F}}{\delta p}(p,\cdot) : \mathcal{X} \to \mathbb{R}$ through

$$\lim_{\varepsilon\to 0} \frac{\mathcal{F}(p+\varepsilon\eta) - \mathcal{F}(p)}{\varepsilon} \triangleq \int \frac{\delta\mathcal{F}}{\delta p}(p,\boldsymbol{x})\, \eta(\mathrm{d}\boldsymbol{x}). \tag{9}$$

Since η integrates to zero, $\frac{\delta\mathcal{F}}{\delta p}$ is defined only up to an additive constant. Under standard regularity and integrability assumptions, this (strictly) convex minimization problem has a unique minimum. Standard variational calculus techniques identify the unique minimizer with the following equation:

$$p^\star(\boldsymbol{x}) = \frac{1}{Z(p^\star)}\, p_{\mathrm{ref}}(\boldsymbol{x}) \exp\left(-\frac{1}{\beta}\frac{\delta\mathcal{F}}{\delta p}(p^\star,\boldsymbol{x})\right), \tag{10}$$

where $Z(p^\star)$ is the normalizing constant. Eq. (10) is generally an implicit fixed point condition because the functional derivative may itself depend on $p^\star$. We refer the reader to Hu et al. (2021) for a formal treatment of this problem, and to Appendix C.1 for a looser derivation.

Expected rewards as the linear case. Standard expected reward maximization, Eq. (2), corresponds to $\mathcal{F}(p) = -\mathbb{E}_{\boldsymbol{x}\sim p}[r(\boldsymbol{x})]$, which is linear in p with first variation $-r(\boldsymbol{x})$. Eq. (10) then reduces to the familiar exponential tilt:

$$p^\star(\boldsymbol{x}) = \frac{1}{Z}\, p_{\mathrm{ref}}(\boldsymbol{x}) \exp\left(\frac{r(\boldsymbol{x})}{\beta}\right), \tag{11}$$

whose right-hand side has no dependence on $p^\star$.

Algorithm 1 Online RWR with DFR. The shaded line is the only change to the baseline.

Require: pretrained velocity v_{ref}; target reward r; distributional functional $\mathcal{R}$; coefficients β, $1/\alpha$, λ; group size G

$\theta \leftarrow \theta_{\text{ref}}, \quad \theta_{\text{old}} \leftarrow \theta_{\text{ref}}$
for each iteration **do**
  for each prompt $\boldsymbol{c}$ **do**
    Sample a group $\{\boldsymbol{x}_0^i\}_{i=1}^G \sim p_{\theta_{\text{old}}}(\cdot \mid \boldsymbol{c})$
    $A^i \leftarrow \big(r(\boldsymbol{x}_0^i, \boldsymbol{c}) - \text{mean}_j\ r(\boldsymbol{x}_0^j, \boldsymbol{c})\big)/\sigma_r$ ▷ group-centered advantage
    $A^i \leftarrow A^i + \lambda\big(r_{\mathcal{R}}(\boldsymbol{x}_0^i; \hat{p}) - \text{mean}_j\ r_{\mathcal{R}}(\boldsymbol{x}_0^j; \hat{p})\big)$ ▷ $r_{\mathcal{R}} = -\delta\mathcal{R}/\delta p$
  end for
  for each sample $(\boldsymbol{x}_0, \boldsymbol{c}, A)$, with $t \sim p(t)$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$ **do**
    $\boldsymbol{x}_t \leftarrow (1-t)\,\boldsymbol{x}_0 + t\,\boldsymbol{\epsilon}, \quad v_{\text{fwd}} \leftarrow \boldsymbol{\epsilon} - \boldsymbol{x}_0$
    $v_{\text{tgt}} \leftarrow \text{sg}\big[v_\theta + \beta\,(v_{\text{ref}} - v_\theta) + \frac{1}{\alpha} A\,(v_{\text{fwd}} - v_{\text{old}})\big]$ ▷ all velocities at $(\boldsymbol{x}_t, t, \boldsymbol{c})$
    Take a gradient step on $\|v_\theta(\boldsymbol{x}_t, t, \boldsymbol{c}) - v_{\text{tgt}}\|_2^2$
  end for
  Update θ_{old} as a lagged (EMA) copy of θ
end for

Distribution-level effective reward. We can add a convex distributional penalty $\mathcal{R}$ to the expected reward as a standard Lagrange multiplier as follows

$$\mathcal{F}(p) = -\mathbb{E}_{\boldsymbol{x}\sim p}[r(\boldsymbol{x})] + \lambda\,\mathcal{R}(p, p_{\text{ref}}), \qquad \lambda \geq 0, \tag{12}$$

where, the functionals derivative defines the distribution-dependent effective reward

$$r_{\text{eff}}(\boldsymbol{x}; p) \triangleq -\frac{\delta\mathcal{F}}{\delta p}(p, \boldsymbol{x}) = r(\boldsymbol{x}) - \lambda\frac{\delta\mathcal{R}}{\delta p}(p, p_{\text{ref}}; \boldsymbol{x}). \tag{13}$$

The optimal distribution therefore satisfies

$$p^\star(\boldsymbol{x}) = \frac{1}{Z(p^\star)} p_{\text{ref}}(\boldsymbol{x}) \exp\left(\frac{r_{\text{eff}}(\boldsymbol{x}; p^\star, p_{\text{ref}})}{\beta}\right). \tag{14}$$

This has the same form as the exponential tilt in Eq. (11), except that the effective reward is evaluated at the optimal distribution itself: each sample's reward depends on the distribution it helps shape, where the right-hand side now has an explicit dependence on $p^\star$.

3.2 Optimizing distributional functionals via policy gradients

Consider a generator p_θ trained to minimize Eq. (12). Using the identity $\nabla_\theta p_\theta = p_\theta \nabla_\theta \log p_\theta$,

$$\nabla_\theta \mathcal{R}(p_\theta, p_{\text{ref}}) = \int \frac{\delta\mathcal{R}}{\delta p}(p_\theta, p_{\text{ref}}; \boldsymbol{x})\,\nabla_\theta p_\theta(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \mathbb{E}_{\boldsymbol{x}\sim p_\theta}\left[\frac{\delta\mathcal{R}}{\delta p}(p_\theta, p_{\text{ref}}; \boldsymbol{x})\,\nabla_\theta \log p_\theta(\boldsymbol{x})\right]. \tag{15}$$

This demonstrates that the gradient of the functional objective is equivalent to an ordinary policy gradient with the effective reward of Eq. (13) in place of the original reward, r, as follows

$$-\nabla_\theta \mathcal{F}(p_\theta) = \mathbb{E}_{\boldsymbol{x}\sim p_\theta}\left[r_{\text{eff}}(\boldsymbol{x}; p_\theta)\,\nabla_\theta \log p_\theta(\boldsymbol{x})\right]. \tag{16}$$

Eq. (16) is exact and independent of the generator. Specifically, any policy-gradient method, whether trajectory-based or ELBO-based, and whether or not the model is flow-based, optimizes $\mathcal{F}$ by substituting r_{eff} for r. Like r, the distributional term is a per-sample scalar evaluated at the current p_θ, so no gradient flows through the distribution penalty, $\mathcal{R}$, or through any feature extractor it uses.

Application to flow models. We use RAM (Bergmeister et al., 2026), the state-of-the-art ELBO-based method, as our base framework. The score in Eq. (16) is replaced by the ELBO surrogate $-\nabla_\theta \mathcal{L}_{\text{FM}}(\theta; \boldsymbol{x})$ alluded to in Eq. (1), and the reverse-KL term in Eq. (8) by $\mathcal{L}_{\text{anchor}}(\theta)$ (Eq. (4)). The complete training pipeline is provided in Algorithm 1.

Algorithm 1 implements the population formulation with three finite-sample choices. All quantities are conditional: $\mathcal{R}$ is applied to $p_\theta(\cdot \mid \boldsymbol{c})$ and averaged over conditions $\boldsymbol{c}$, which we omit for

brevity. The distributional reward is centered within each group and added to the normalized task advantage rather than to r before normalization; the two differ only in the rescaling of λ by the advantage scale. The population statistics entering $r_{\mathcal{R}}$ are estimated from the group of G samples for MMD and DreamSim diversity, excluding the sample itself from pairwise averages, and from an exponential moving average of feature moments for FD (detailed in Appendix C.6.3).

3.3 Instantiations: MMD, FD, DreamSim Diversity

We instantiate $\mathcal{R}$ with three functionals: (1) the MMD reward, our primary choice, which requires no pretrained encoder; (2) the FD reward; and (3) the DreamSim reward. The latter two show that DFR subsumes recently proposed diversity and FD-based rewards as special cases; Section 4 evaluates all three under the same protocol. All three functionals are convex in p (Appendix C.4), so the analysis in Section 3.1 applies directly.

MMD reward. Let $\psi : \mathcal{X} \to \mathbb{R}^m$ be a fixed injective map and $b \in (0, 2)$, and define

$$\rho_b(\boldsymbol{x}, \boldsymbol{y}) = \|\psi(\boldsymbol{x}) - \psi(\boldsymbol{y})\|_2^b. \tag{17}$$

We use the identity map for ψ, meaning ρ_b reduces to a power of the Euclidean distance in latent space and requires no pretrained encoder, although an encoder could be used instead. The corresponding MMD can be written as

$$\mathrm{MMD}^2_{k_b}(p, p_{\text{ref}}) = \mathbb{E}_{\substack{\boldsymbol{x}\sim p \\ \boldsymbol{y}\sim p_{\text{ref}}}} [\rho_b(\boldsymbol{x}, \boldsymbol{y})] - \frac{1}{2}\mathbb{E}_{\boldsymbol{x},\boldsymbol{x}'\sim p} [\rho_b(\boldsymbol{x}, \boldsymbol{x}')] - \frac{1}{2}\mathbb{E}_{\boldsymbol{y},\boldsymbol{y}'\sim p_{\text{ref}}} [\rho_b(\boldsymbol{y}, \boldsymbol{y}')] . \tag{18}$$

The proof is in Appendix C.3. Consider the MMD functional $\mathcal{R}_{\mathrm{MMD}}(p, p_{\text{ref}}) = \mathrm{MMD}^2_{k_b}(p, p_{\text{ref}})$.

Proposition 1 (MMD reward)**.** *A representative of the negative functional derivative of $\mathcal{R}_{\mathrm{MMD}}$ is*

$$\boxed{r_{\mathrm{MMD}}(\boldsymbol{x}; p, p_{ref}) = \mathbb{E}_{\boldsymbol{x}'\sim p} [\rho_b(\boldsymbol{x}, \boldsymbol{x}')] - \mathbb{E}_{\boldsymbol{y}\sim p_{ref}} [\rho_b(\boldsymbol{x}, \boldsymbol{y})] .} \tag{19}$$

The proof is given in Appendix C.5. The two terms in Eq. (19) have complementary roles. The first is a repulsive term that rewards samples far from other samples generated by the current model, thereby discouraging collapse. The second is an attractive term that penalizes samples far from the pretrained distribution, thereby limiting distributional drift.

FD reward. DFR also recovers FD-based reward augmentation. With μ_p, Σ_p as in Section 2.3, consider the FD functional $\mathcal{R}_{\mathrm{FD}}(p) = \mathrm{FD}(p, p_{\text{ref}})$ of Eq. (5).

Proposition 2 (FD reward)**.** *Assume the covariance matrices are positive definite and define*

$$G_p = I - \Sigma_q^{1/2} \left(\Sigma_q^{1/2}\Sigma_p\Sigma_q^{1/2}\right)^{-1/2} \Sigma_q^{1/2} \tag{20}$$

for $q \equiv p_{ref}$. A representative of the negative functional derivative of $\mathcal{R}_{\mathrm{FD}}$ is

$$\boxed{r_{\mathrm{FD}}(\boldsymbol{x}; p) = -\left[2(\mu_p - \mu_q)^\top f + (f - \mu_p)^\top G_p (f - \mu_p)\right] .} \tag{21}$$

The proof is given in Appendix C.6. The two terms in Eq. (21) have complementary roles. The first penalizes shifts in the population feature mean, while the second corrects changes in feature covariance.

DreamSim diversity reward. DFR also recovers pairwise perceptual diversity rewards. With $z(\boldsymbol{x})$ defined in Section 2.3, the cosine similarity between two samples is $z(\boldsymbol{x})^\top z(\boldsymbol{x}')$. We consider the population-average cosine-similarity functional $\mathcal{R}_{\mathrm{DS}}(p) = \frac{1}{2}(1 - \mathrm{Div}_{\mathrm{DS}}(p))$ of Eq. (7).

Proposition 3 (DreamSim reward)**.** *A representative of the negative functional derivative of $\mathcal{R}_{\mathrm{DS}}$ is*

$$\boxed{r_{\mathrm{DS}}(\boldsymbol{x}; p) = -z(\boldsymbol{x})^\top \mathbb{E}_{\boldsymbol{x}'\sim p}[z(\boldsymbol{x}')].} \tag{22}$$

The proof is given in Appendix C.7.

4 Experiments

We evaluate whether DFR improves the trade-off between reward and reward hacking beyond what tuning the reverse-KL coefficient of the baseline can achieve.

Table 1: Fine-tuning on GenEval, compared at matched GenEval (Section 4.1). Peak is the highest threshold it reaches or, if none, its mean GenEval at the evaluated checkpoints; "–" marks a threshold not reached on every seed. Speedup is the time to reach 0.95 relative to Flow-GRPO. Entries of the form a/b report results on the GenEval test prompts (a) and on DrawBench (b); RAM and DFR entries are means over three seeds with SD beneath, and existing methods are single checkpoints evaluated at 0.95. Blue marks a DFR entry that wins against every RAM configuration reaching the same threshold; struck-through entries are computed with an encoder related to that instance's reward and are not compared.

			GenEval 0.95			**GenEval** 0.96		
Method	**Peak**	**Speedup** (↑)	**FID** (↓)	**FD-DINOv2** (↓)	**DreamSim** (↑)	**FID** (↓)	**FD-DINOv2** (↓)	**DreamSim** (↑)
SD3.5-M	0.63	–	–	–	0.302/0.274			
Trajectory-based								
Flow-GRPO		1.0	114.32/125.61	1019/997	0.155/0.189			
RSM[§]		6.25	132.82/154.29	1271/1310	0.142/0.183			
ELBO-based								
AWM*		8.0	158.88/183.71	1412/1578	0.095/0.132			
DiffusionNFT		24.0	144.36/159.22	1352/1437	0.109/0.143			
DiffusionNFT (multi)[‡]		2.28	145.06/175.20	1468/1702	0.102/0.106			
RAM ($\beta = 1$)	0.95	52.3	132.93/144.77 ±16.19/16.96	1211/1236 ±116/185	0.163/0.193 ±0.021/0.021	–	–	–
RAM with larger β								
RAM ($\beta = 2$)	0.96	57.2 ±6.1	116.99/122.78 ±1.69/1.02	1078/1002 ±34/11	0.170/0.208 ±0.009/0.006	119.87/125.99 ±2.03/3.89	1066/1023 ±31/33	0.174/0.213 ±0.006/0.006
RAM ($\beta = 3$)	0.95	43.9 ±4.9	113.19/122.06 ±2.83/1.84	1030/991 ±11/9	0.184/0.214 ±0.006/0.010	–	–	–
RAM ($\beta = 4$)	0.94	–	–	–	–	–	–	–
Ours								
DFR-MMD ($\beta = 2$)	0.96	41.1 ±9.1	111.20/120.64 ±4.10/6.17	990/967 ±27/52	0.200/0.224 ±0.015/0.011	110.75/121.67 ±3.04/6.05	992/977 ±5/47	0.196/0.221 ±0.011/0.009
DFR-FD ($\beta = 2$)	0.96	35.7 ±4.6	111.02/117.09 ±2.00/4.88	980/932 ±20/46	0.199/0.231 ±0.003/0.006	111.95/120.40 ±4.11/4.05	994/969 ±46/38	0.193/0.223 ±0.011/0.013
DFR-DS ($\beta = 2$)	0.96	40.4 ±5.2	114.60/121.86 ±1.64/4.22	~~1034~~/~~992~~ ±~~37/82~~	0.218/0.233 ±0.007/0.010	114.47/124.11 ±0.31/3.88	~~1052~~/~~1007~~ ±~~8/45~~	0.225/0.240 ±0.022/0.018

[§]RSM jumps from 0.94 to 0.97 between consecutive checkpoints, so we evaluate it at 0.94.
*AWM's default configuration uses a $1.5\times$ larger effective batch size per step; we match the batch size to the other methods.
[‡]Released checkpoint (GenEval 0.94), sampled at its prescribed guidance scale of 1.0.

4.1 Setup

Tasks and methods. We fine-tune Stable Diffusion 3.5 Medium (SD3.5-M; Esser et al., 2024) on two tasks: (1) Compositional generation (GenEval; Ghosh et al., 2023), with reward thresholds 0.95 and 0.96, and (2) Visual text rendering (OCR), with threshold 0.96. We adopt RAM (Bergmeister et al., 2026) as the ELBO-based framework on which DFR builds on. We sweep its reverse-KL coefficient β over $\{1, \ldots, 4\}$ for GenEval and $\{2, \ldots, 6\}$ for OCR. DFR-MMD, DFR-FD, and DFR-DS augment RAM with the distributional rewards introduced in Section 3.3 (Algorithm 1). For each variant, we use the largest β for which RAM reaches the highest reward threshold, or the next smaller value when the distributional reward itself provides substantial regularization toward the pretrained model. All configurations are trained with three random seeds under identical compute budgets.

Baselines. We compare DFR against other ELBO- and trajectory-based methods including DiffusionNFT (Zheng et al., 2026), AWM (Xue et al., 2026), RSM (Lee et al., 2026), and Flow-GRPO (Liu et al., 2025b) on both the compositional generation and visual text rendering tasks.

Protocol. Because reward hacking intensifies over the course of optimization, we compare models at matched target reward levels. For each seed, we evaluate the first checkpoint that reaches a given threshold; a configuration is considered to reach that threshold only if all three seeds do so.

Metrics. Following Liu et al. (2025d;c); Domingo-Enrich et al. (2025), we measure reward hacking by drift and diversity, each computed per-prompt over 64 images. *Drift* is the Fréchet distance to the 64 images of p_{ref} in Inception-v3 and DINOv2 features. *Diversity* is measured by DreamSim diversity, computed with its DINO and OpenCLIP branches. Preference rewards such as ImageReward (Xu et al., 2023) and PickScore (Kirstain et al., 2023) are reported but not used, since they score each image in isolation and cannot detect a model that generates near-identical images for every seed. Two configurations are tied on a metric if their means differ by at most the larger of their standard deviations; otherwise, the configuration with the better mean wins. Entries computed with an encoder that a DFR instance uses in its reward, or a closely related one, are counted as neither wins nor losses. Consequently, the main text reports the OpenCLIP branch, which no instance trains on. Additional training, evaluation, and baseline details are provided in Appendix D.

Table 2: Fine-tuning on OCR, compared at OCR reward 0.96. Columns and markings are as in Table 1, except that entries report results on the OCR test prompts (a) and on DrawBench (b), and Time is the time to reach 0.96 relative to RAM ($\beta = 2$).

			OCR 0.96		
Method	**Peak**	**Time** (↓)	**FID** (↓)	**FD-DINOv2** (↓)	**DreamSim** (↑)
SD3.5-M	0.59	–	–	–	0.258/0.274
RAM ($\beta = 2$)	0.96	1.00	219.07/131.38 ±24.04/7.71	1270/1040 ±232/84	0.128/0.245 ±0.010/0.009
RAM ($\beta = 3$)	0.95	–	–	–	–
RAM ($\beta = 4$)	0.96	1.85 ±0.23	138.04/100.94 ±6.73/4.36	667/751 ±46/56	0.185/0.249 ±0.004/0.005
RAM ($\beta = 5$)	0.96	2.05 ±0.09	136.78/97.16 ±5.35/2.14	662/721 ±39/21	0.192/0.252 ±0.005/0.003
RAM ($\beta = 6$)	0.95	–	–	–	–
DFR-MMD ($\beta = 4$)	0.96	2.50 ±0.19	124.87/95.37 ±0.83/1.60	597/708 ±5/16	0.197/0.261 ±0.006/0.008
DFR-FD ($\beta = 4$)	0.96	2.70 ±0.34	121.46/94.56 ±4.52/1.20	585/705 ±20/13	0.213/0.255 ±0.004/0.008
DFR-DS ($\beta = 5$)	0.96	2.08 ±0.22	136.13/99.23 ±4.59/3.20	~~653/743~~ ~~±37/37~~	0.224/0.281 ±0.008/0.003

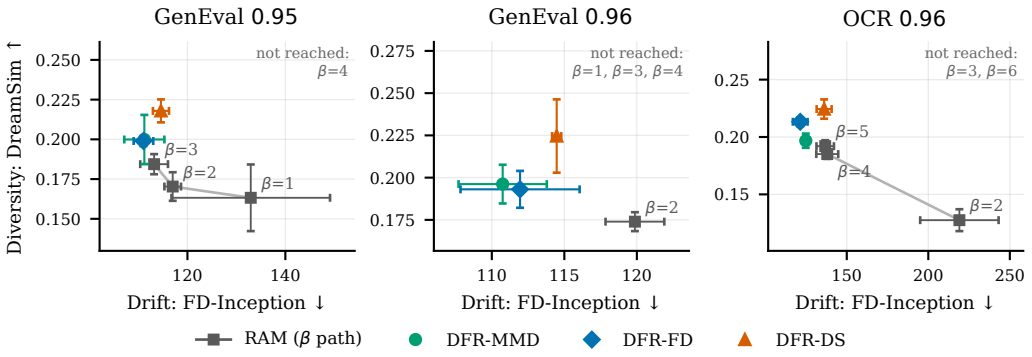


Figure 2: Drift versus diversity at matched reward. Tuning β moves RAM along a path (grey) that ends where a larger β no longer reaches the threshold. Every DFR instance lies above or to the left of the end of the path.

4.2 Results

DFR moves beyond the β trade-off. Every DFR instance lies beyond RAM's β path, with less drift and/or more diversity (Figure 2). Further, DFR also wins or ties against all RAM configurations that reaches the same threshold on every drift and diversity metric (Tables 1 and 2). The gap is widest at the highest threshold, GenEval 0.96: among RAM configurations only $\beta = 2$ reaches it, whereas all three DFR instances do, with lower drift (FID 111–114 vs. 120) and higher diversity (0.193–0.225 vs. 0.174). On OCR, DFR-MMD and DFR-FD drift the least, and DFR-DS is the most diverse.

The instances behave as their rewards predict. The DreamSim reward is purely repulsive, whereas MMD and FD rewards also pull samples toward p_{ref} (Section 3.3). Accordingly, on every task and threshold, DFR-DS attains the highest diversity of the three instances but the smallest reduction in drift, while DFR-MMD and DFR-FD reduce drift the most. On OCR, for example, DFR-DS matches RAM ($\beta = 5$) in FID (136 vs. 137), while raising diversity by 17%, whereas DFR-MMD and DFR-FD lower FID to 121–125 despite running at the smaller $\beta = 4$. The attractive term also acts as additional regularization: on OCR, DFR-DS trains as fast as RAM at the same β, whereas DFR-MMD and DFR-FD require a smaller β to reach the threshold and still take longer.

DFR-MMD matches DFR-FD without an encoder. DFR-MMD computes its reward on SD3.5-M's own VAE latents with eight reference samples per prompt, whereas DFR-FD relies on three pretrained encoders, precomputed reference statistics, and a per-encoder calibration. Despite this, DFR-MMD ties DFR-FD on every drift metric at every threshold, matches its diversity on GenEval, and reaches the thresholds at least as fast, although it is less diverse on OCR (0.197 vs. 0.213).

DFR adds modest cost. Distributional rewards add only 3–9% to the time per training step. Reaching a threshold takes longer than RAM at the same β, but the cost is comparable to that of reducing drift by raising β. Image quality and prompt generalization are also preserved: ImageReward and PickScore remain comparable to those of the RAM configurations that reach the same threshold.

5 Related Work

Regularization for reward fine-tuning. Most reward fine-tuning methods for flow-based models use the velocity anchor in Eq. (4), which constrains the sampling dynamics rather than p_θ itself (Section 3.1). DDRL (Ye et al., 2025) instead regularizes toward an off-policy data distribution using the forward-KL divergence, implemented through an ELBO objective on real data or pretrained-model samples. In contrast, DFR directly regularizes the generated distribution through functionals of p_θ.

Diversity and distributional rewards. Several works augment the target reward with diversity or distributional signals. GARDO (He et al., 2025) increases the rewards of high-quality samples far from their nearest neighbors, while DiverseGRPO (Liu et al., 2025a) assigns bonuses based on semantic-cluster sizes; neither signal is derived from an explicit distributional objective. DRIFT (Liu et al., 2026) rewards each sample by its mean squared DreamSim distance to the other samples in its group, which coincides with the DreamSim instance of DFR (see Appendix C.7).

Other methods match a prescribed target distribution. LOO-FID (Baran et al., 2026) and Miao et al. (2024) reward samples by their leave-one-out change in, respectively, the FD to real-image statistics and the MMD to curated reference images; these are finite-difference estimates of the corresponding first variations (Appendix C.6.2). kCGM (Diamant & Trippe, 2026) minimizes MMD to a target feature distribution using an estimator of the form in Eq. (15). DistWise (Li et al., 2026) uses the FID between a large generated set and real data as a distribution-level reward, while DRAGON (Bai et al., 2025) contrasts positive and negative generation sets under distribution-level rewards. In these methods, matching a real or curated target distribution—typically in a pretrained feature space—is itself the objective. By contrast, DFR derives per-sample rewards as first variations of convex functionals of p_θ, yielding the explicit objective in Eq. (12), where deviation from the pretrained model distribution regularizes target-reward maximization alongside the KL anchor. Consequently, DFR requires neither an image dataset nor, for its MMD instance, a pretrained encoder.

6 Conclusion

In this work, we introduce Distributional Functional Rewards (DFR), a general-purpose framework for reward fine-tuning that complements KL regularization with objectives defined directly on the model's output distribution. By deriving per-sample rewards from the first variations of distributional functionals, DFR provides explicit control over diversity and distributional drift. Our findings clearly demonstrate that KL regularization has a limited effective range: beyond it, stronger regularization slows optimization, limits reward improvement, and/or destabilizes training. On two distinct tasks: compositional generation and visual text rendering, we find that DFR consistently improves the reward–distribution trade-off beyond what is attainable by tuning the KL coefficient alone. At matched task reward, every evaluated DFR variant improves diversity, reduces drift, or both, while matching or improving all eligible distributional metrics relative to KL-only RAM baselines that reach the same reward threshold. These gains require only a simple reward augmentation, adding 3–9% overhead per training step, while not relying on a pretrained encoder. Overall, DFR enables reward fine-tuning to achieve high task rewards, while better preserving diversity and remaining closer to the pretrained model distribution than KL regularization alone.

Limitations and future work. Although presented as a general-purpose reward fine-tuning framework, our experiments are focused on validation in image settings; extending DFR to other modalities such as video and molecular structures is a natural next step to more effectively evaluate the generalization of the framework. Further, since the distributional reward regularizes toward p_{ref}, it preserves the pretrained model's biases along with its diversity, so its coefficient λ must be tuned jointly with β. Balancing DFR's repulsive and attractive terms more carefully is a promising direction for improving this trade-off. Our characterization of the optimum assumes exact optimization of the population objective, which the ELBO surrogate and the finite-sample estimates of the distributional rewards approximate in practice.

AI USE STATEMENT

In this work, we used generative AI tools to help improve the text, math, and related work; provide assistance with methodological implementation; and suggest feedback on the research conducted. All AI-assisted work was judiciously verified by all authors, and we take full responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Our work develops a general-purpose framework for reward fine-tuning of flow-based models. Our experiments focus on text-to-image generation using standard research datasets and pretrained models. As with other post-training methods, the framework could be adopted to optimize models toward arbitrary reward functions, including objectives that amplify biases or facilitate the generation of harmful or misleading content. We encourage the use of appropriate safeguards, carefully-designed rewards, and evaluation for unintended behaviors when applying the approach to real-world systems.

REPRODUCIBILITY STATEMENT

We provide implementation details, hyperparameters, and theoretical proofs in the appendix. All datasets used are publicly available. We will release our code in a public repository upon publication.

REFERENCES

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J. Ballard, Joshua Bambrick, and et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, May 2024. doi: 10.1038/s41586-024-07487-w.

Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint 2303.08797*, 2023.

Yatong Bai, Jonah Casebeer, Somayeh Sojoudi, and Nicholas J. Bryan. DRAGON: Distributional rewards optimize diffusion generative models. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=gobhDku03J.

Orhun Bugra Baran, Melih Kandemir, and Ramazan Gokberk Cinbis. Policy-based tuning of autoregressive image models with instance- and distribution-level rewards, 2026. URL https://arxiv.org/abs/2603.23086.

Andreas Bergmeister, Stefanie Jegelka, Nikolas Nüsken, Carles Domingo-Enrich, and Jakiw Pidstrigach. Reinforce adjoint matching: Scaling rl post-training of diffusion and flow-matching models, 2026. URL https://arxiv.org/abs/2605.10759.

R. Bhatia. *Matrix Analysis*. Graduate Texts in Mathematics. Springer New York, 1996. ISBN 9780387948461. URL https://books.google.ca/books?id=F4hRy1F1M6QC.

Mikołaj Bińkowski, Dougal J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=r1lUOzWCW.

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

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

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

Valentin De Bortoli, Alexandre Galashov, J. Swaroop Guntupalli, Guangyao Zhou, Kevin Murphy, Arthur Gretton, and Arnaud Doucet. Distributional diffusion models with scoring rules. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

Nathaniel L. Diamant and Brian L. Trippe. Calibrating generative models to feature distributions with mmd finetuning, 2026. URL https://arxiv.org/abs/2606.19496.

Carles Domingo-Enrich, Michal Drozdzal, Brian Karrer, and Ricky T. Q. Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=xQBRrtQM8u.

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. *arXiv preprint arXiv:2403.03206*, 2024. URL https://arxiv.org/abs/2403.03206.

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

Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Reinforcement learning for fine-tuning text-to-image diffusion models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=8OTPepXzeh.

Dan Friedman and Adji Bousso Dieng. The vendi score: A diversity evaluation metric for machine learning, 2023. URL https://arxiv.org/abs/2210.02410.

Stephanie Fu, Netanel Yakir Tamir, Shobhita Sundaram, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. Dreamsim: Learning new dimensions of human visual similarity using synthetic data. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=DEiNSfh1k7.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. GenEval: An object-focused framework for evaluating text-to-image alignment. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=Wbr51vK331.

Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. A kernel method for the two-sample-problem. In B. Schölkopf, J. Platt, and T. Hoffman (eds.), *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2006. URL https://proceedings.neurips.cc/paper_files/paper/2006/file/e9fb2eda3d9c55a0d89c98d6c54b5b3e-Paper.pdf.

Haoran He, Yuxiao Ye, Jie Liu, Jiajun Liang, Zhiyong Wang, Ziyang Yuan, Xintao Wang, Hangyu Mao, Pengfei Wan, and Ling Pan. Gardo: Reinforcing diffusion models without reward hacking, 2025. URL https://arxiv.org/abs/2512.24138.

Xiaoxuan He, Siming Fu, Yuke Zhao, Wanli Li, Jian Yang, Dacheng Yin, Fengyun Rao, and Bo Zhang. TempFlow-GRPO: When timing matters for GRPO in flow models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=7mCo3R3Wyn.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

Kaitong Hu, Zhenjie Ren, David Šiška, and Łukasz Szpruch. Mean-field langevin dynamics and energy landscape of neural networks. *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, 57(4):2043–2065, 2021. doi: 10.1214/20-AIHP1140.

Nai-Chieh Huang, Ping-Chun Hsieh, Kuo-Hao Ho, and I-Chen Wu. PPO-Clip attains global optimality: Towards deeper understandings of clipping. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 12600–12607, 2024. doi: 10.1609/aaai.v38i11.29154. URL https://ojs.aaai.org/index.php/AAAI/article/view/29154.

Sadeep Jayasumana, Srikumar Ramalingam, Andreas Veit, Daniel Glasner, Ayan Chakrabarti, and Sanjiv Kumar. Rethinking fid: Towards a better evaluation metric for image generation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9307–9315, 2024. doi: 10.1109/CVPR52733.2024.00889.

Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: an open dataset of user preferences for text-to-image generation. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Branislav Kveton, Anup Rao, Subhojyoti Mukherjee, Krishna Kumar Singh, and Viet Dac Lai. Advantageflow: Advantage-weighted least squares for rl in flow models, 2026. URL https://arxiv.org/abs/2605.26013.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/0234c510bc6d908b28c70ff313743079-Paper.pdf.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space, 2025. URL https://arxiv.org/abs/2506.15742.

Jeongjae Lee and Jong Chul Ye. PCPO: Proportionate credit policy optimization for preference alignment of image generation models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=alY08iknli.

Jeongjae Lee, Jinho Chang, Jeongsol Kim, and Jong Chul Ye. Reward score matching: Unifying reward-based fine-tuning for flow and diffusion models, 2026. URL https://arxiv.org/abs/2604.17415.

Junzhe Li, Yutao Cui, Tao Huang, Yinping Ma, Chun Fan, Miles Yang, and Zhao Zhong. MixGRPO: Unlocking flow-based GRPO efficiency with mixed ODE-SDE, 2025. URL https://arxiv.org/abs/2507.21802.

Ruihang Li, Mengde Xu, Shuyang Gu, Leigang Qu, Fuli Feng, Han Hu, and Wenjie Wang. Optimizing visual generative models via distribution-wise rewards. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=dxi8kGgKMM.

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

Henglin Liu, Huijuan Huang, Jing Wang, Chang Liu, Xiu Li, and Xiangyang Ji. Diversegrpo: Mitigating mode collapse in image generation via diversity-aware grpo, 2025a. URL https://arxiv.org/abs/2512.21514.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-GRPO: Training flow matching models via online RL. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b. URL https://openreview.net/forum?id=oCBKGw5HNf.

Jinmei Liu, Haoru Li, Zhenhong Sun, Chaofeng Chen, Yatao Bian, Bo Wang, Daoyi Dong, Chunlin Chen, and Zhi Wang. Beyond the dirac delta: Mitigating diversity collapse in reinforcement fine-tuning for versatile image generation, 2026. URL https://arxiv.org/abs/2601.12401.

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

Zhen Liu, Tim Z. Xiao, Carles Domingo-Enrich, Weiyang Liu, and Dinghuai Zhang. Value gradient guidance for flow matching alignment. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025c. URL https://openreview.net/forum?id=6MmOy2Ji8V.

Zhen Liu, Tim Z. Xiao, Weiyang Liu, Yoshua Bengio, and Dinghuai Zhang. Efficient diversity-preserving diffusion alignment via gradient-informed GFlownets. In *The Thirteenth International Conference on Learning Representations*, 2025d. URL https://openreview.net/forum?id=Aye5wL6TCn.

Weijian Luo, Tianyang Hu, Shifeng Zhang, Jiacheng Sun, Zhenguo Li, and Zhihua Zhang. Diff-instruct: A universal approach for transferring knowledge from pre-trained diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

David McAllister, Songwei Ge, Brent Yi, Chung Min Kim, Ethan Weber, Hongsuk Choi, Haiwen Feng, and Angjoo Kanazawa. Flow matching policy gradients. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=eoEmoKoQpJ.

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

Stefano Peluchetti. Non-denoising forward-time diffusions. *arXiv preprint arXiv:2312.14589*, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

Tim Salimans, Thomas Mensink, Jonathan Heek, and Emiel Hoogeboom. Multistep distillation of diffusion models via moment matching, 2024. URL https://arxiv.org/abs/2406.04103.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

Dino Sejdinovic, Bharath Sriperumbudur, Arthur Gretton, and Kenji Fukumizu. Equivalence of distance-based and RKHS-based statistics in hypothesis testing. *The Annals of Statistics*, 41 (5):2263 – 2291, 2013. doi: 10.1214/13-AOS1140. URL https://doi.org/10.1214/13-AOS1140.

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

Jeongwoo Shin, Dongsoo Shin, Yuchen Zhu, Wei Guo, Yongxin Chen, Joonseok Lee, Jaewoong Choi, and Jaemoo Choi. Efficient adjoint matching for fine-tuning diffusion models, 2026. URL https://arxiv.org/abs/2605.11480.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2256–2265, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

George Stein, Jesse Cresswell, Rasa Hosseinzadeh, Yi Sui, Brendan Ross, Valentin Villecroze, Zhaoyan Liu, Anthony L Caterini, Eric Taylor, and Gabriel Loaiza-Ganem. Exposing flaws of generative model evaluation metrics and their unfair treatment of diffusion models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 3732–3784. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/0bc795afae289ed465a65a3b4b1f4eb7-Paper-Conference.pdf.

Alexander Tong, Nikolay Malkin, Kilian Fatras, Lazar Atanackovic, Yanlei Zhang, Guillaume Huguet, Guy Wolf, and Yoshua Bengio. Simulation-free schr "odinger bridges via score and flow matching, July 2023.

Masatoshi Uehara, Yulai Zhao, Tommaso Biancalani, and Sergey Levine. Understanding reinforcement learning-based fine-tuning of diffusion models: A tutorial and review, 2024. URL https://arxiv.org/abs/2407.13734.

Masatoshi Uehara, Yulai Zhao, Chenyu Wang, Xiner Li, Aviv Regev, Sergey Levine, and Tommaso Biancalani. Inference-time alignment in diffusion models with reward-guided generation: Tutorial and review, 2025. URL https://arxiv.org/abs/2501.09685.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models, 2025. URL https://arxiv.org/abs/2503.20314.

R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. ImageReward: learning and evaluating human preferences for text-to-image generation. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pp. 15903–15935, 2023.

Yilun Xu, Weili Nie, and Arash Vahdat. One-step diffusion models with f-divergence distribution matching, 2025. URL https://arxiv.org/abs/2502.15681.

Yixian Xu, Yuanrui Zhang, Shengjie Luo, Liwei Wang, and Di He. Designing reinforcement learning for diffusion models: A unified path-space view, 2026. URL https://arxiv.org/abs/2608.14430.

Shuchen Xue, Chongjian GE, Shilong Zhang, Yichen Li, and Zhi-Ming Ma. Advantage weighted matching: Aligning RL with pretraining in diffusion models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=nLY2pOYBrJ.

Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, and Ping Luo. DanceGRPO: Unleashing grpo on visual generation, 2025. URL https://arxiv.org/abs/2505.07818.

Jiawei Yang, Zhengyang Geng, Xuan Ju, Yonglong Tian, and Yue Wang. Representation fréchet loss for visual generation, 2026. URL https://arxiv.org/abs/2604.28190.

Haotian Ye, Kaiwen Zheng, Jiashu Xu, Puheng Li, Huayu Chen, Jiaqi Han, Sheng Liu, Qinsheng Zhang, Hanzi Mao, Zekun Hao, Prithvijit Chattopadhyay, Dinghao Yang, Liang Feng, Maosheng Liao, Junjie Bai, Ming-Yu Liu, James Zou, and Stefano Ermon. Data-regularized reinforcement learning for diffusion models at scale, 2025. URL https://arxiv.org/abs/2512.04332.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *CVPR*, 2024a.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T. Freeman. Improved distribution matching distillation for fast image synthesis, 2024b. URL https://arxiv.org/abs/2405.14867.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su, Stefano Ermon, Jun Zhu, and Ming-Yu Liu. DiffusionNFT: Online diffusion reinforcement with forward process. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=VJZ477R89F.

Linqi Zhou, Stefano Ermon, and Jiaming Song. Inductive moment matching. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=pwNSUo7yUb.

A Extended Related Work

Differentiable rewards. When the reward is differentiable, its gradient can be propagated directly through the generative process. DRaFT (Clark et al., 2024) backpropagates through the sampling chain, optionally truncating the gradient to the final steps, whereas ReFL (Xu et al., 2023) backpropagates through a single late denoising step from which the clean sample is predicted directly. Adjoint Matching (Domingo-Enrich et al., 2025) formulates fine-tuning as stochastic optimal control under a memoryless noise schedule, whose optimum solves Eq. (2), and regresses the velocity toward a target defined by an adjoint state that propagates the reward gradient backward along the trajectory. Shin et al. (2026) reduce its cost using a linear base drift, enabling few-step deterministic sampling and a closed-form adjoint. VGG-Flow (Liu et al., 2025c) matches the difference between fine-tuned and pretrained velocities to the gradient of a learned value function, while ∇-GFlowNet (Liu et al., 2025d) derives gradient-informed GFlowNet objectives; both seek to preserve the pretrained prior and sample diversity. For broader unifying perspectives on reward fine-tuning, we refer readers to Uehara et al. (2024); Choi et al. (2026); Lee et al. (2026); Xu et al. (2026).

Non-differentiable rewards. Trajectory-based methods formulate denoising as a multi-step MDP and apply policy gradients to per-step transition likelihoods (Black et al., 2024; Fan et al., 2023; Liu et al., 2025b; Xue et al., 2025). Because these likelihoods require stochastic transitions, the sampling ODE is converted to an SDE with the same marginals. Li et al. (2025); He et al. (2026) improve efficiency by restricting stochastic sampling and optimization to a subset of timesteps.

ELBO-based methods instead treat the final sample as a single action and replace its intractable log-likelihood $\log p_\theta(\boldsymbol{x}_0)$ with the ELBO surrogate $-\mathcal{L}_{\text{FM}}(\theta)$, producing an online reward-weighted regression objective on forward-noised samples (McAllister et al., 2026; Fan et al., 2025; Xue et al., 2026). Although this gives a biased estimator of the exact policy gradient, it is empirically effective. Methods derived from other perspectives reduce to the same objective (Zheng et al., 2026; Bergmeister et al., 2026); Appendix B gives these reductions in a common notation. By decoupling optimization from sampling, ELBO-based methods can substantially accelerate reward maximization relative to trajectory-based methods (Choi et al., 2026).

Distributional discrepancies as training losses. Several few-step distillation methods match the output distribution of a student generator to that of a pretrained teacher, through the integral reverse-KL divergence (Luo et al., 2024; Yin et al., 2024a;b), general f-divergences (Xu et al., 2025), or moment matching (Salimans et al., 2024). Other works match the data distribution directly. Zhou et al. (2025) train few-step generators from scratch with an MMD-based moment-matching objective, and De Bortoli et al. (2025) learn the full conditional distribution of clean data given a noisy state with scoring rules such as the energy score, rather than regressing its mean. FD-loss (Yang et al., 2026) post-trains generators by minimizing the Fréchet distance to real data in a representation space.

A.1 Measuring fidelity and diversity

Distribution-level fidelity. FID (Heusel et al., 2017) compares generated and real images through the Fréchet distance between Gaussian fits to their Inception-v3 features. Its limitations have motivated several alternatives. Stein et al. (2023) find that Inception features align poorly with human judgments of modern generators and recommend DINOv2 features instead. KID (Bińkowski et al., 2018) and CMMD (Jayasumana et al., 2024) replace FD with MMD, which drops the Gaussian assumption and admits an unbiased estimator. Precision and recall (Kynkäänniemi et al., 2019) separate fidelity from coverage by estimating the support of each distribution with k-nearest-neighbor balls in feature space.

Diversity. LPIPS (Zhang et al., 2018) and DreamSim (Fu et al., 2023) compare images through deep features calibrated to human similarity judgments. LPIPS mainly reflects low-level differences such as color and texture, whereas DreamSim also captures layout, pose, and semantic content. The mean pairwise perceptual distance within a set of samples is a common diversity measure in reward fine-tuning (Domingo-Enrich et al., 2025; Liu et al., 2025d;c). The Vendi Score (Friedman & Dieng, 2023) instead measures the effective number of distinct samples, as the exponential of the entropy of the eigenvalues of the normalized similarity matrix.

B UNIFYING ONLINE RWR

Several recent ELBO-based methods have different motivations, but their updates reduce to the same online reward-weighted regression (RWR) objective (Lee et al., 2026; Kveton et al., 2026; Choi et al., 2026). This appendix derives the reduction for ORW-CFM-W2, FPO, AWM, RAM, DiffusionNFT, and AdvantageFlow, and summarizes it in Table 3.

Notation. We translate every method into the conventions of Section 2.1 and Algorithm 1. Time $t = 0$ is clean data and $t = 1$ is noise [2] , with $\boldsymbol{x}_t = (1-t)\,\boldsymbol{x}_0 + t\,\boldsymbol{\epsilon}$, $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$, and $v_{\text{fwd}} = \boldsymbol{\epsilon} - \boldsymbol{x}_0$. We write v_θ, v_{old}, and v_{ref} for the velocities of the current, lagged, and pretrained models, all evaluated at $(\boldsymbol{x}_t, t)$, and omit conditioning. Rollouts are drawn as $\boldsymbol{x}_0 \sim p_{\theta_{\text{old}}}$, with $\theta_{\text{old}} = \theta$ for on-policy methods, and timesteps are sampled from a distribution $p(t)$, i.e., $t \sim p(t)$. As in the main text, β is the coefficient of the anchor to v_{ref}; we write γ for the coefficient of the anchor to v_{old}. The reward enters through a scalar weight $f(r(\boldsymbol{x}_0), t)$. For most methods $f = A(\boldsymbol{x}_0)/\alpha$, where A is the group-normalized advantage and $1/\alpha$ a reward coefficient; for ORW-CFM-W2, $f = \exp(r(\boldsymbol{x}_0)/\alpha)$.

Unified gradient. Setting aside clipping and loss transformations, which we discuss for each method, every method below updates θ along

$$g(\theta) = \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{\epsilon}}\Big[2(\nabla_\theta v_\theta)^\top \Big(\underbrace{\beta\,(v_\theta - v_{\text{ref}})}_{\text{ref. anchor}} + \underbrace{\gamma\,(v_\theta - v_{\text{old}})}_{\text{old anchor}} + \underbrace{f(r(\boldsymbol{x}_0), t)\,(v_\bullet - v_{\text{fwd}})}_{\text{RWR}} \Big)\Big], \qquad (23)$$

$$v_\bullet \in \{v_\theta, v_{\text{old}}\},$$

where the sampling distribution of $\boldsymbol{x}_0$ is held fixed when differentiating. Eq. (23) is exactly the gradient of the regression loss

$$\mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{\epsilon}}\Big\| v_\theta - \text{sg}\Big[v_\theta + \beta\,(v_{\text{ref}} - v_\theta) + \gamma\,(v_{\text{old}} - v_\theta) + f(r(\boldsymbol{x}_0), t)\,(v_{\text{fwd}} - v_\bullet)\Big]\Big\|_2^2, \qquad (24)$$

since the stop-gradient leaves only $2(\nabla_\theta v_\theta)^\top (v_\theta - \text{sg}[\cdot])$. For $v_\bullet = v_\theta$ it is also exactly the gradient of the sum of squares $\mathbb{E}\big[f\,\|v_\theta - v_{\text{fwd}}\|_2^2 + \beta\|v_\theta - v_{\text{ref}}\|_2^2 + \gamma\|v_\theta - v_{\text{old}}\|_2^2\big]$, in which the weight f does not depend on θ.

We call two objectives equivalent when their gradients agree up to a positive constant factor, which can be absorbed into the learning rate. A recurring case is several methods that multiply every term of their loss by the same positive time weight $w(t)$. We absorb this weight into the time distribution, replacing $\mathbb{E}_{t\sim p}[w(t)\,h(t)]$ by $\mathbb{E}_{t\sim\tilde{p}}[h(t)]$, where $\tilde{p}(t) = p(t)\,w(t)/Z$ and $Z = \mathbb{E}_{t\sim p}[w(t)]$. Since Z does not depend on θ, the two are equivalent. Finally, some reductions hold exactly only at $\theta = \theta_{\text{old}}$, i.e., for on-policy updates; these are marked with an asterisk in Table 3.

Table 3: Online RWR methods as instances of the unified gradient, Eq. (23). Methods marked with * reduce exactly only at $\theta = \theta_{\text{old}}$ (Appendix B.6); all other reductions are exact identities. Time weights are relative to $t \sim \mathcal{U}(0,1)$: sampling $t \sim p(t)$ is equivalent to uniform sampling with weight $p(t)$.

Method	β	γ	f	$v_\bullet$	Time weight	Details
ORW-CFM-W2	their α	0	$\exp(r/\alpha)$	v_θ	1	App. B.1
FPO*	0	0	A	v_θ	$\tilde{w}_{\text{FPO}}(t)$	App. B.2
AWM*,†	their β	0	A	v_θ	$w_{\text{AWM}}(t)$	App. B.3
RAM§	1	0	A/α	v_{old}	$2t$	App. B.4
DiffusionNFT	0‡	1	$\tilde{A}/\zeta$	v_{old}	1	App. B.5
AdvantageFlow	their λ	1.1 or $1-\tilde{A}$	$\tilde{A}$	v_θ	t^2	App. B.5
Algorithm 1 (ours)	β	0	$(A + \lambda\,\bar{r}_{\mathcal{R}})/\alpha$	v_{old}	$2t$	Sec. 3.2

†Follows its Equations 13, 15, and 16; its main experiments anchor to an EMA of the online model instead.
§Follows its official implementation; its Equation 17 has $v_\bullet = v_\theta$.
‡Its official implementation adds a small reference anchor, i.e., $\beta > 0$.

[2] ORW-CFM-W2 and FPO denote clean data at time 1; we substitute $t \mapsto 1 - t$.

B.1 ORW-CFM-W2

ORW-CFM-W2 (Fan et al., 2025) minimizes a reward-weighted flow-matching loss plus an anchor to the pretrained velocity (Fan et al., 2025, Equation 10). Their reward weight $\exp(\tau\, r)$ and anchor coefficient α correspond to our $f = \exp(r(\boldsymbol{x}_0)/\alpha)$ and β, and rollouts are drawn from the current model, i.e., $\theta_{\text{old}} = \theta$. Their loss is therefore exactly the sum-of-squares form below Eq. (24), with $\gamma = 0$ and $v_\bullet = v_\theta$ (Table 3).

B.2 FPO

For each rollout $\boldsymbol{x}_0 \sim p_{\theta_{\text{old}}}$ with advantage $A = A(\boldsymbol{x}_0)$, FPO draws N_{mc} pairs $(t_i, \boldsymbol{\epsilon}_i)$ with $t_i \sim \mathcal{U}(0,1)$ and evaluates the $\boldsymbol{\epsilon}$-prediction loss $\ell_\theta(t_i, \boldsymbol{\epsilon}_i) = w_{\text{FPO}}(t_i)\, \|\hat{\boldsymbol{\epsilon}}_\theta(\boldsymbol{x}_{t_i}, t_i) - \boldsymbol{\epsilon}_i\|_2^2$, where $w_{\text{FPO}}(t) > 0$ is the weighting of McAllister et al. (2026, Equation 14). Rewriting the model's prediction as $\hat{\boldsymbol{\epsilon}}_\theta = \boldsymbol{x}_t + (1-t)\, v_\theta$ gives $\hat{\boldsymbol{\epsilon}}_\theta - \boldsymbol{\epsilon} = (1-t)(v_\theta - v_{\text{fwd}})$, and hence

$$\ell_\theta(t, \boldsymbol{\epsilon}) = \tilde{w}_{\text{FPO}}(t)\, \|v_\theta - v_{\text{fwd}}\|_2^2, \qquad \tilde{w}_{\text{FPO}}(t) = w_{\text{FPO}}(t)\, (1-t)^2. \tag{25}$$

The ratio and the clipped objective are (McAllister et al., 2026, Algorithm 1, lines 9–10)

$$\hat{\rho}_\theta = \exp\Big(-\frac{1}{N_{\text{mc}}} \sum_{i=1}^{N_{\text{mc}}} \big[\ell_\theta(t_i, \boldsymbol{\epsilon}_i) - \ell_{\theta_{\text{old}}}(t_i, \boldsymbol{\epsilon}_i)\big]\Big), \qquad \mathcal{L}_{\text{FPO}}(\theta) = -\min\big(\hat{\rho}_\theta A,\ \text{clip}(\hat{\rho}_\theta, 1-\xi, 1+\xi)\, A\big), \tag{26}$$

where we negate FPO's objective to obtain a loss to be minimized.

At $\theta = \theta_{\text{old}}$, $\hat{\rho}_\theta = 1$ lies strictly inside $[1-\xi, 1+\xi]$, so no rollout is clipped, and $\nabla_\theta \hat{\rho}_\theta = \hat{\rho}_\theta \nabla_\theta \log \hat{\rho}_\theta = \nabla_\theta \log \hat{\rho}_\theta$. The gradient of Eq. (26) is therefore

$$\nabla_\theta \mathcal{L}_{\text{FPO}}(\theta_{\text{old}}) = -A\, \nabla_\theta \log \hat{\rho}_\theta = \frac{1}{N_{\text{mc}}} \sum_{i=1}^{N_{\text{mc}}} 2\, \tilde{w}_{\text{FPO}}(t_i)\, (\nabla_\theta v_\theta)^\top A\, (v_\theta - v_{\text{fwd}}), \tag{27}$$

with the velocities in the i-th term evaluated at $(\boldsymbol{x}_{t_i}, t_i)$. Averaging over $(t_i, \boldsymbol{\epsilon}_i)$ and absorbing $\tilde{w}_{\text{FPO}}(t)$ into the time distribution, this is exactly Eq. (23) with $\beta = \gamma = 0$, $f = A$, $v_\bullet = v_\theta$, and $p(t) \propto \tilde{w}_{\text{FPO}}(t)$. For $\theta \neq \theta_{\text{old}}$, the importance ratio and clipping modify this gradient (Appendix B.6).

B.3 AWM

AWM (Xue et al., 2026) uses the same importance ratio and clipping as FPO (Appendix B.6), but with a velocity-prediction loss and an additional anchor to the pretrained model. The regression loss entering the ratio is (Xue et al., 2026, Equation 15)

$$\ell_\theta(t, \boldsymbol{\epsilon}) = w_{\text{AWM}}(t)\, \|v_\theta - v_{\text{fwd}}\|_2^2, \tag{28}$$

averaged over N_{mc} pairs $(t_i, \boldsymbol{\epsilon}_i)$, and the anchor is $\beta\, w_{\text{AWM}}(t)\, \|v_\theta - v_{\text{ref}}\|_2^2$, whose coefficient coincides with our β (Xue et al., 2026, Equations 13 and 16). With PPO clipping, which the paper omits from its algorithm for simplicity but its implementation includes, the full loss is

$$\mathcal{L}_{\text{AWM}}(\theta) = -\min\big(\hat{\rho}_\theta A,\ \text{clip}(\hat{\rho}_\theta, 1-\xi, 1+\xi)\, A\big) + \frac{\beta}{N_{\text{mc}}} \sum_{i=1}^{N_{\text{mc}}} w_{\text{AWM}}(t_i)\, \|v_\theta - v_{\text{ref}}\|_2^2, \tag{29}$$

with $\hat{\rho}_\theta$ as in Eq. (36) and $\bar{\ell}_\theta$ the average of Eq. (28).

As for FPO, at $\theta = \theta_{\text{old}}$ we have $\hat{\rho}_\theta = 1$, no rollout is clipped, and the ratio term contributes $-A\, \nabla_\theta \log \hat{\rho}_\theta = A\, \nabla_\theta \bar{\ell}_\theta$. Hence

$$\nabla_\theta \mathcal{L}_{\text{AWM}}(\theta_{\text{old}}) = \frac{1}{N_{\text{mc}}} \sum_{i=1}^{N_{\text{mc}}} 2\, w_{\text{AWM}}(t_i)\, (\nabla_\theta v_\theta)^\top \big[A\, (v_\theta - v_{\text{fwd}}) + \beta\, (v_\theta - v_{\text{ref}})\big], \tag{30}$$

with the velocities in the i-th term evaluated at $(\boldsymbol{x}_{t_i}, t_i)$. The weight $w_{\text{AWM}}(t)$ multiplies both terms, so absorbing it into the time distribution, this is exactly Eq. (23) with $\gamma = 0$, $f = A$, $v_\bullet = v_\theta$, and $p(t) \propto w_{\text{AWM}}(t)$. For $\theta \neq \theta_{\text{old}}$, see Appendix B.6.

Caveat. In its public implementation[3], AWM replaces the loss inside the ratio, Eq. (28), by a concave transform of the squared error, which it calls a generalized Huber loss:

$$\ell_\theta(t,\boldsymbol{\epsilon}) = \frac{t\left[\left(\frac{1}{D}\|v_\theta - v_{\text{fwd}}\|_2^2 + \delta\right)^{1/2} - \delta^{1/2}\right]}{1/2}, \qquad \delta = 10^{-10}, \tag{31}$$

where D is the data dimension. This transform is not described in the paper.

B.4 RAM

RAM (Bergmeister et al., 2026) derives its objective from stochastic optimal control (SOC), but approximates the path-cost correction term by zero; this approximation is what reduces the objective to reward-weighted regression. Note that without the correction, the SOC formulation is poorly justified. Therefore, we instead analyze the objective RAM optimizes in practice (Bergmeister et al., 2026, Equation 17),

$$\mathcal{L}_{\text{RAM}}(\theta) = \mathbb{E}_{t\sim p(t),\, \boldsymbol{x}_0\sim p_{\theta_{\text{old}}},\boldsymbol{\epsilon}}\left\|v_\theta - \text{sg}\big[v_{\text{ref}} + A\,(v_{\text{fwd}} - v_\theta)\big]\right\|_2^2, \qquad p(t) = 2t, \tag{32}$$

where we write A for the group-normalized advantage that RAM denotes by r. With $\beta = 1$, the terms v_θ and $-\beta\, v_\theta$ inside the stop-gradient of Eq. (24) cancel, so Eq. (32) is exactly Eq. (24) with $\beta = 1$, $\gamma = 0$, $f = A$, and $v_\bullet = v_\theta$.

In its implementation, RAM makes two changes: it scales the advantage by a reward coefficient $1/\alpha$, and evaluates the velocity inside the RWR term at the lagged parameters, using v_{old} in place of v_θ. The implemented loss, $\mathbb{E}\left\|v_\theta - \text{sg}\big[v_{\text{ref}} + \frac{A}{\alpha}(v_{\text{fwd}} - v_{\text{old}})\big]\right\|_2^2$, is therefore Eq. (24) with $\beta = 1$, $\gamma = 0$, $f = A/\alpha$, and $v_\bullet = v_{\text{old}}$. Note that this is equivalent to Algorithm 1 with $\beta = 1$ and $\lambda = 0$.

B.5 DiffusionNFT & AdvantageFlow

DiffusionNFT. With clipped advantage $\tilde{A} = \text{clip}(A, -1, 1)$, the loss is

$$\mathcal{L}_{\text{NFT}}(\theta) = \mathbb{E}_{t,\boldsymbol{x}_0\sim p_{\theta_{\text{old}}},\boldsymbol{\epsilon}}\left[\tfrac{1+\tilde{A}}{2}\left\|(1-\zeta)\,v_{\text{old}} + \zeta\, v_\theta - v_{\text{fwd}}\right\|_2^2 + \tfrac{1-\tilde{A}}{2}\left\|(1+\zeta)\,v_{\text{old}} - \zeta\, v_\theta - v_{\text{fwd}}\right\|_2^2\right], \tag{33}$$

where we have rewritten their β with $\zeta(\in \{1, 0.1\})$ (Zheng et al., 2026, Equation 5). Lee et al. (2026, Appendix H.2.1) shows that Eq. (33) is exactly ζ^2 times Eq. (24) with $\beta = 0$, $\gamma = 1$, $f = \tilde{A}/\zeta$, and $v_\bullet = v_{\text{old}}$, up to a θ-independent constant. Their public implementation additionally adds a small anchor to the pretrained velocity, $\beta_{\text{NFT}}\|v_\theta - v_{\text{ref}}\|_2^2$ with default $\beta_{\text{NFT}} = 10^{-4}$, which is not part of the paper; it corresponds to some $\beta > 0$ in Eq. (24), whose exact value depends on the implementation's loss scaling and on the adaptive weighting discussed below[4].

AdvantageFlow (Kveton et al., 2026, Equation 5) states its objective in terms of the clean-sample prediction $\hat{\boldsymbol{x}}_0 = \boldsymbol{x}_t - t\, v$. Since $\hat{\boldsymbol{x}}_0 - \boldsymbol{x}_0 = t\,(v - v_{\text{fwd}})$, every term carries a factor t^2, which we absorb into the time distribution $p(t)$. The resulting objective is equivalent to Eq. (24) with $f = \tilde{A}$ and $v_\bullet = v_\theta$. AdvantageFlow sets γ so that the total coefficient $\tilde{A} + \gamma + \beta$ is positive for every sample, e.g., $\gamma = 1.1$ or $\gamma = 1 - \tilde{A}$, which makes each per-sample loss strictly convex in the prediction (Kveton et al., 2026, Section 3.3).

Caveat. DiffusionNFT replaces the time weighting of the flow-matching loss by an adaptive self-normalization, which it motivates as an alternative to manual tuning of the weighting (Zheng et al., 2026, page 6). Each of the two squared errors in Eq. (33) is divided by the stop-gradient of its mean absolute $\boldsymbol{x}_0$-prediction error. Since $\hat{\boldsymbol{x}}_0 = \boldsymbol{x}_t - t\, v$ gives $\hat{\boldsymbol{x}}_0 - \boldsymbol{x}_0 = t\,(v - v_{\text{fwd}})$, the first and second terms acquire the weights

$$\omega_1 = \frac{t}{\text{sg}\big(\text{mean}\big|(1-\zeta)\,v_{\text{old}} + \zeta\, v_\theta - v_{\text{fwd}}\big|\big)}, \qquad \omega_2 = \frac{t}{\text{sg}\big(\text{mean}\big|(1+\zeta)\,v_{\text{old}} - \zeta\, v_\theta - v_{\text{fwd}}\big|\big)}. \tag{34}$$

[3] AWM official implementation

[4] DiffusionNFT official implementation

At $\theta = \theta_{\text{old}}$, both residuals equal $v_{\text{old}} - v_{\text{fwd}}$, so $\omega_1 = \omega_2$ and the gradient is that of Eq. (33) multiplied by a per-sample factor. Away from θ_{old}, however, $\omega_1 \neq \omega_2$, and the two terms no longer combine into Eq. (24). Differentiating the weighted terms directly, the coefficient of the RWR direction $v_{\text{old}} - v_{\text{fwd}}$ becomes

$$\tilde{f} = \tfrac{\zeta}{2}\big[(1+\tilde{A})\,\omega_1 - (1-\tilde{A})\,\omega_2\big], \tag{35}$$

which is no longer proportional to $\tilde{A}$: a rollout with $\tilde{A} = 0$ still receives the weight $\zeta(\omega_1 - \omega_2)/2$. The adaptive weighting therefore changes the scale *and update direction*.

B.6 Importance Sampling ratios and clipping

FPO and AWM weight each rollout's regression loss by an estimated importance ratio and clip it as in PPO (Schulman et al., 2017). We describe the effect for a rollout $\boldsymbol{x}_0$ with advantage A. Let $\bar{\ell}_\theta$ be its regression loss averaged over N_{mc} pairs $(t_i, \boldsymbol{\epsilon}_i)$, e.g., the average of Eq. (25), and

$$\hat{\rho}_\theta = \exp\big(-(\bar{\ell}_\theta - \bar{\ell}_{\theta_{\text{old}}})\big), \qquad \mathcal{L}(\theta) = -\min\big(\hat{\rho}_\theta A,\ \text{clip}(\hat{\rho}_\theta, 1-\xi, 1+\xi)\,A\big). \tag{36}$$

Since $\bar{\ell}_\theta$ is only an ELBO surrogate for $-\log p_\theta(\boldsymbol{x}_0)$, up to a θ-independent constant, $\hat{\rho}_\theta$ estimates the likelihood ratio $p_\theta(\boldsymbol{x}_0)/p_{\theta_{\text{old}}}(\boldsymbol{x}_0)$ rather than equaling it.

Clipping. If $A > 0$ and $\hat{\rho}_\theta > 1+\xi$, or $A < 0$ and $\hat{\rho}_\theta < 1-\xi$, the minimum in Eq. (36) selects the clipped branch, which is constant in θ, and the rollout contributes no gradient. In the other two cases the minimum selects $\hat{\rho}_\theta A$. Clipping thus stops updating a rollout once its ratio has moved past $1 \pm \xi$ in the direction favored by A, a trust region around θ_{old}.

Importance sampling ratio. When the rollout is not clipped, using $\nabla_\theta \hat{\rho}_\theta = \hat{\rho}_\theta \nabla_\theta \log \hat{\rho}_\theta$,

$$\nabla_\theta \mathcal{L}(\theta) = -A\,\hat{\rho}_\theta\,\nabla_\theta \log \hat{\rho}_\theta = \hat{\rho}_\theta\,A\,\nabla_\theta \bar{\ell}_\theta, \tag{37}$$

i.e., the RWR gradient $A\,\nabla_\theta \bar{\ell}_\theta$ multiplied by the per-rollout factor $\hat{\rho}_\theta$. At $\theta = \theta_{\text{old}}$, $\hat{\rho}_\theta = 1$ and no rollout is clipped, which is why the reductions of FPO and AWM hold exactly there. Away from θ_{old}, the step is reweighted by $\hat{\rho}_\theta$ and truncated by clipping, and Eq. (23) captures neither.

Log-ratio surrogate. Huang et al. (2024) show that PPO clipping is a hinge loss on $\rho - 1$, and that $\rho - 1$ can be replaced by other functions that change sign at $\rho = 1$, such as $\log \rho$; PCPO (Lee & Ye, 2026) reports that $\widehat{\log \rho}$ is a lower-variance estimator than $\hat{\rho}$ for ρ for diffusion models. Replacing $\hat{\rho}_\theta$ by $\log \hat{\rho}_\theta + 1$ in Eq. (36) turns the unclipped gradient into $-A\,\nabla_\theta \log \hat{\rho}_\theta = A\,\nabla_\theta \bar{\ell}_\theta$ for every θ, removing the multiplicative factor $\hat{\rho}_\theta$. Clipping $\log \hat{\rho}_\theta + 1$ to $[1-\xi, 1+\xi]$ is the same as clipping $\log \hat{\rho}_\theta$ to $[-\xi, \xi]$, so the trust region remains.

C Distributional rewards: proofs and details

Throughout this appendix, we interchangeably use $q \equiv p_{\text{ref}}$ for brevity.

C.1 Informal derivation of the first-order condition

We derive Eq. (10) formally, assuming $p^\star$ has a density that is positive wherever p_{ref} is, and that the functional derivative of Eq. (9) exists; Hu et al. (2021, Proposition 2.5) give a rigorous version under conditions on $\mathcal{F}$ and p_{ref}.

Write $\mathcal{V}(p) = \mathcal{F}(p) + \beta\,\mathcal{D}_{\text{KL}}(p\|p_{\text{ref}})$. The functional derivative of the KL term is

$$\frac{\delta}{\delta p}\,\mathcal{D}_{\text{KL}}(p\|p_{\text{ref}})(\boldsymbol{x}) = \log\frac{p(\boldsymbol{x})}{p_{\text{ref}}(\boldsymbol{x})} + 1, \tag{38}$$

since, writing the KL divergence as $\int p \log(p/p_{\text{ref}})\,\mathrm{d}\boldsymbol{x}$ and differentiating under the integral,

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathcal{D}_{\text{KL}}(p+\varepsilon\eta\|p_{\text{ref}})\Big|_{\varepsilon=0} = \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\int \big(p(\boldsymbol{x})+\varepsilon\eta(\boldsymbol{x})\big)\log\frac{p(\boldsymbol{x})+\varepsilon\eta(\boldsymbol{x})}{p_{\text{ref}}(\boldsymbol{x})}\,\mathrm{d}\boldsymbol{x}\,\Big|_{\varepsilon=0} \tag{39}$$

$$= \int \frac{\partial}{\partial u}\Big[u\log\frac{u}{p_{\text{ref}}(\boldsymbol{x})}\Big]_{u=p(\boldsymbol{x})} \cdot \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\big(p(\boldsymbol{x})+\varepsilon\eta(\boldsymbol{x})\big)\,\mathrm{d}\boldsymbol{x} \tag{40}$$

$$= \int \Big(\log\frac{p(\boldsymbol{x})}{p_{\text{ref}}(\boldsymbol{x})} + 1\Big)\,\eta(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}. \tag{41}$$

If $p^\star$ minimizes $\mathcal{V}$, then for every zero-mass η such that $p^\star + \varepsilon\eta$ remains a probability measure for small ε of either sign,

$$0 = \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathcal{V}(p^\star + \varepsilon\eta)\Big|_{\varepsilon=0} = \int \left[\frac{\delta\mathcal{F}}{\delta p}(p^\star, \boldsymbol{x}) + \beta \log \frac{p^\star(\boldsymbol{x})}{p_{\text{ref}}(\boldsymbol{x})}\right] \eta(\mathrm{d}\boldsymbol{x}), \tag{42}$$

where the $+1$ of Eq. (38) has dropped out because $\eta(\mathcal{X}) = 0$. A function whose integral against every zero-mass η vanishes is constant, so

$$\frac{\delta\mathcal{F}}{\delta p}(p^\star, \boldsymbol{x}) + \beta \log \frac{p^\star(\boldsymbol{x})}{p_{\text{ref}}(\boldsymbol{x})} = c \qquad \text{for all } \boldsymbol{x}, \tag{43}$$

and solving for $p^\star$ gives $p^\star(\boldsymbol{x}) \propto p_{\text{ref}}(\boldsymbol{x}) \exp\!\big(-\frac{1}{\beta}\frac{\delta\mathcal{F}}{\delta p}(p^\star, \boldsymbol{x})\big)$, with the constant c absorbed into $Z(p^\star)$. Convexity of $\mathcal{F}$ and strict convexity of the KL term make $\mathcal{V}$ strictly convex; therefore, the stationary point is the unique minimizer.

C.2 Backward consistency.

By the chain rule for relative entropy,

$$\mathcal{D}_{\mathrm{KL}}\big(\mathbb{P}\|\mathbb{P}_{\text{ref}}\big) = \mathcal{D}_{\mathrm{KL}}\big(\mathbb{P}_0\|p_{\text{ref}}\big) + \mathbb{E}_{x_0\sim\mathbb{P}_0}\left[\mathcal{D}_{\mathrm{KL}}(\mathbb{P}(\mathrm{d}\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0)\|\mathbb{P}_{\text{ref}}(\mathrm{d}\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0))\right]. \tag{44}$$

Because $\mathcal{F}$ depends only on the clean-endpoint marginal, the conditional KL is minimized by

$$\mathbb{P}^\star(\mathrm{d}\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0) = \mathbb{P}_{\text{ref}}(\mathrm{d}\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0). \tag{45}$$

The optimal process therefore changes the distribution of clean samples, while preserving the pretrained conditional noising process. Equivalently, its backward kernels coincide with those of the pretrained model. This posterior-preservation property was established by Uehara et al. (2024, Theorem 3) and underlies forward-process fine-tuning methods like DiffusionNFT (Zheng et al., 2026).

C.3 Energy distance as an MMD

Following the energy score construction of De Bortoli et al. (2025), we construct the kernel behind $\mathcal{R}_{\mathrm{MMD}}$ and show that its squared MMD equals Eq. (18). Let $\psi : \mathcal{X} \to \mathbb{R}^m$ be a fixed injective map and $b \in (0, 2)$, and define

$$\rho_b(\boldsymbol{x}, \boldsymbol{y}) = \|\psi(\boldsymbol{x}) - \psi(\boldsymbol{y})\|_2^b. \tag{46}$$

Consider distributions with a finite b-moment, $\mathbb{E}_{\boldsymbol{x}\sim p}\|\psi(\boldsymbol{x})\|_2^b < \infty$, so that every expectation of ρ_b is finite. The function ρ_b is of negative type (Sejdinovic et al., 2013, Proposition 3) and induces the positive definite distance-induced kernel

$$k_b(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2}\left[\rho_b(\boldsymbol{x}, \boldsymbol{x}_{\mathrm{a}}) + \rho_b(\boldsymbol{y}, \boldsymbol{x}_{\mathrm{a}}) - \rho_b(\boldsymbol{x}, \boldsymbol{y})\right], \tag{47}$$

where $\boldsymbol{x}_{\mathrm{a}} \in \mathcal{X}$ is an arbitrary anchor (Sejdinovic et al., 2013, Section 4.1). The squared MMD does not depend on the choice of $\boldsymbol{x}_{\mathrm{a}}$ and equals half of the energy distance (Sejdinovic et al., 2013, Theorem 22), given as Eq. (18).

C.4 Convexity of the distributional functionals

Section 3.1 requires $\mathcal{R}$ to be convex along mixtures: for $p_t = (1-t)p_0 + tp_1$ and $t \in [0, 1]$, $\mathcal{R}(p_t) \le (1-t)\mathcal{R}(p_0) + t\mathcal{R}(p_1)$. We verify this for the three instances of Section 3.3.

MMD. Let $\nu = p - q$. By Eq. (6), $\mathcal{R}_{\mathrm{MMD}}(p) = Q(\nu) := \iint k_b(\boldsymbol{x}, \boldsymbol{y})\,\nu(\mathrm{d}\boldsymbol{x})\,\nu(\mathrm{d}\boldsymbol{y})$, a quadratic form in the signed measure ν. Because k_b is positive definite, $Q(\nu) \ge 0$ for every signed measure ν with finite b-moment. Since $p \mapsto \nu$ is affine, $\nu_t = (1-t)\nu_0 + t\nu_1$, and expanding the quadratic form,

$$Q(\nu_t) = (1-t)Q(\nu_0) + tQ(\nu_1) - t(1-t)\,Q(\nu_0 - \nu_1) \le (1-t)Q(\nu_0) + tQ(\nu_1). \tag{48}$$

DreamSim diversity. Let $m(p) = \mathbb{E}_p[z] \in \mathbb{R}^d$, so that $\mathcal{R}_{\mathrm{DS}}(p) = \frac{1}{2}\|m(p)\|_2^2$. Since $m(p_t) = (1-t)m(p_0) + t\,m(p_1)$, and for any m_0, m_1,

$$\tfrac{1}{2}\|(1-t)m_0 + t\,m_1\|_2^2 = (1-t)\tfrac{1}{2}\|m_0\|_2^2 + t\,\tfrac{1}{2}\|m_1\|_2^2 - t(1-t)\tfrac{1}{2}\|m_0 - m_1\|_2^2 \tag{49}$$

$$\le (1-t)\tfrac{1}{2}\|m_0\|_2^2 + t\,\tfrac{1}{2}\|m_1\|_2^2, \tag{50}$$

$\mathcal{R}_{\mathrm{DS}}$ is convex.

FD. Expanding the two p-dependent terms of Eq. (5) separately,

$$\|\mu_p - \mu_q\|_2^2 = \|\mu_p\|_2^2 - 2\mu_p^\top \mu_q + \|\mu_q\|_2^2, \tag{51}$$

$$\operatorname{Tr}\Sigma_p = \operatorname{Tr}\big(\mathbb{E}_p[ff^\top]\big) - \operatorname{Tr}\big(\mu_p\mu_p^\top\big) = \mathbb{E}_p\|f\|_2^2 - \|\mu_p\|_2^2, \tag{52}$$

where the last equality uses $\operatorname{Tr}(vv^\top) = \|v\|_2^2$. Adding the two, the $\|\mu_p\|_2^2$ terms cancel:

$$\|\mu_p - \mu_q\|_2^2 + \operatorname{Tr}\Sigma_p = \mathbb{E}_p\|f\|_2^2 - 2\mu_p^\top \mu_q + \|\mu_q\|_2^2. \tag{53}$$

Both $\mathbb{E}_p\|f\|_2^2$ and $\mu_p = \mathbb{E}_p[f]$ are expectations under p and hence linear in p, so this sum is affine in p. Substituting into Eq. (5) and writing $g(A) := \operatorname{Tr}\big(\Sigma_q^{1/2} A\, \Sigma_q^{1/2}\big)^{1/2}$,

$$\mathcal{R}_{\mathrm{FD}}(p) = \underbrace{\mathbb{E}_p\|f\|_2^2 - 2\mu_p^\top \mu_q + \|\mu_q\|_2^2 + \operatorname{Tr}\Sigma_q}_{\text{affine in } p} - 2\, g(\Sigma_p). \tag{54}$$

It remains to show that $p \mapsto g(\Sigma_p)$ is concave.

First, we show that $p \mapsto \Sigma_p$ is concave in the Loewner order, i.e., $\Sigma_{p_t} \succeq (1-t)\Sigma_{p_0} + t\,\Sigma_{p_1}$. To see this, write $M_i = \mathbb{E}_{p_i}[ff^\top]$ and $\mu_i = \mu_{p_i}$ for $i \in \{0,1\}$, so that $\Sigma_{p_i} = M_i - \mu_i\mu_i^\top$. Expectations are linear in p, so under the mixture

$$\mathbb{E}_{p_t}[ff^\top] = (1-t)M_0 + t\,M_1, \qquad \mu_{p_t} = (1-t)\mu_0 + t\,\mu_1, \tag{55}$$

and therefore

$$\Sigma_{p_t} = (1-t)M_0 + t\,M_1 - \mu_{p_t}\mu_{p_t}^\top. \tag{56}$$

The mixture of the covariances is

$$(1-t)\Sigma_{p_0} + t\,\Sigma_{p_1} = (1-t)M_0 + t\,M_1 - (1-t)\mu_0\mu_0^\top - t\,\mu_1\mu_1^\top. \tag{57}$$

The M terms coincide, so the difference is

$$\Sigma_{p_t} - \big[(1-t)\Sigma_{p_0} + t\,\Sigma_{p_1}\big] = (1-t)\mu_0\mu_0^\top + t\,\mu_1\mu_1^\top - \mu_{p_t}\mu_{p_t}^\top. \tag{58}$$

Expanding the last term,

$$\mu_{p_t}\mu_{p_t}^\top = (1-t)^2\mu_0\mu_0^\top + t^2\mu_1\mu_1^\top + t(1-t)\big(\mu_0\mu_1^\top + \mu_1\mu_0^\top\big), \tag{59}$$

and substituting into Eq. (58) with $(1-t) - (1-t)^2 = t(1-t)$ and $t - t^2 = t(1-t)$,

$$\Sigma_{p_t} - \big[(1-t)\Sigma_{p_0} + t\,\Sigma_{p_1}\big] = t(1-t)\mu_0\mu_0^\top + t(1-t)\mu_1\mu_1^\top - t(1-t)\big(\mu_0\mu_1^\top + \mu_1\mu_0^\top\big) \tag{60}$$

$$= t(1-t)(\mu_0 - \mu_1)(\mu_0 - \mu_1)^\top. \tag{61}$$

Since $t(1-t) \geq 0$ and $vv^\top$ is positive semidefinite for any vector v, the right-hand side is positive semidefinite.

Second, g is nondecreasing and concave on positive semidefinite matrices: $A \mapsto \Sigma_q^{1/2} A \Sigma_q^{1/2}$ is linear and order preserving, the matrix square root is operator monotone and operator concave (Bhatia, 1996, Theorem V.1.9), and the trace is linear and order preserving. Hence

$$g(\Sigma_{p_t}) \geq g\big((1-t)\Sigma_{p_0} + t\Sigma_{p_1}\big) \geq (1-t)\, g(\Sigma_{p_0}) + t\, g(\Sigma_{p_1}), \tag{62}$$

using monotonicity with Eq. (61) and then concavity. Thus $-2g(\Sigma_p)$ is convex, and $\mathcal{R}_{\mathrm{FD}}$ is convex.

C.5 MMD REWARD

Proposition 1 (MMD reward)**.** *A representative of the negative functional derivative of $\mathcal{R}_{\mathrm{MMD}}$ is*

$$\boxed{r_{\mathrm{MMD}}(\boldsymbol{x}; p, p_{ref}) = \mathbb{E}_{\boldsymbol{x}'\sim p}\left[\rho_b(\boldsymbol{x}, \boldsymbol{x}')\right] - \mathbb{E}_{\boldsymbol{y}\sim p_{ref}}\left[\rho_b(\boldsymbol{x}, \boldsymbol{y})\right].} \tag{19}$$

We use $q \equiv p_{\text{ref}}$. Also, assume p, q have finite b-moments, so that every expectation is finite [5].

Name the three terms of Eq. (18):

$$T_1(p) = \iint \rho(\boldsymbol{x}, \boldsymbol{y})\, p(\mathrm{d}\boldsymbol{x})\, q(\mathrm{d}\boldsymbol{y}), \tag{63}$$

$$T_2(p) = \iint \rho(\boldsymbol{x}, \boldsymbol{x}')\, p(\mathrm{d}\boldsymbol{x})\, p(\mathrm{d}\boldsymbol{x}'), \tag{64}$$

$$T_3 = \iint \rho(\boldsymbol{y}, \boldsymbol{y}')\, q(\mathrm{d}\boldsymbol{y})\, q(\mathrm{d}\boldsymbol{y}'), \tag{65}$$

so that $\mathcal{R}_{\mathrm{MMD}}(p) = T_1(p) - \frac{1}{2}T_2(p) - \frac{1}{2}T_3$, with T_3 independent of p. Let η be a signed measure with $\eta(\mathcal{X}) = 0$ and replace p by $p + \varepsilon\eta$.

T_1 is linear in p:

$$T_1(p + \varepsilon\eta) = T_1(p) + \varepsilon \int \Big[\int \rho(\boldsymbol{x}, \boldsymbol{y})\, q(\mathrm{d}\boldsymbol{y})\Big] \eta(\mathrm{d}\boldsymbol{x}) = T_1(p) + \varepsilon \int \mathbb{E}_{\boldsymbol{y}\sim q}\, \rho(\boldsymbol{x}, \boldsymbol{y})\ \eta(\mathrm{d}\boldsymbol{x}). \tag{66}$$

T_2 is quadratic in p. Expanding the product $(p + \varepsilon\eta)(p + \varepsilon\eta)$,

$$\begin{aligned} T_2(p + \varepsilon\eta) &= T_2(p) + \varepsilon \iint \rho(\boldsymbol{x}, \boldsymbol{x}')\, \eta(\mathrm{d}\boldsymbol{x})\, p(\mathrm{d}\boldsymbol{x}') + \varepsilon \iint \rho(\boldsymbol{x}, \boldsymbol{x}')\, p(\mathrm{d}\boldsymbol{x})\, \eta(\mathrm{d}\boldsymbol{x}') \\ &\quad + \varepsilon^2 \iint \rho(\boldsymbol{x}, \boldsymbol{x}')\, \eta(\mathrm{d}\boldsymbol{x})\, \eta(\mathrm{d}\boldsymbol{x}') \qquad (67) \\ &= T_2(p) + 2\varepsilon \int \Big[\int \rho(\boldsymbol{x}, \boldsymbol{x}')\, p(\mathrm{d}\boldsymbol{x}')\Big] \eta(\mathrm{d}\boldsymbol{x}) + O(\varepsilon^2) \qquad (68) \\ &= T_2(p) + 2\varepsilon \int \mathbb{E}_{\boldsymbol{x}'\sim p}\, \rho(\boldsymbol{x}, \boldsymbol{x}')\ \eta(\mathrm{d}\boldsymbol{x}) + O(\varepsilon^2), \qquad (69) \end{aligned}$$

where we use the symmetry $\rho(\boldsymbol{x}, \boldsymbol{x}') = \rho(\boldsymbol{x}', \boldsymbol{x})$.

Combining, and dividing by ε,

$$\lim_{\varepsilon\to 0} \frac{\mathcal{R}_{\mathrm{MMD}}(p + \varepsilon\eta) - \mathcal{R}_{\mathrm{MMD}}(p)}{\varepsilon} = \int \Big[\mathbb{E}_{\boldsymbol{y}\sim q}\, \rho(\boldsymbol{x}, \boldsymbol{y}) - \mathbb{E}_{\boldsymbol{x}'\sim p}\, \rho(\boldsymbol{x}, \boldsymbol{x}')\Big] \eta(\mathrm{d}\boldsymbol{x}). \tag{70}$$

Comparing with Eq. (9), a representative of the functional derivative is

$$\frac{\delta \mathcal{R}_{\mathrm{MMD}}}{\delta p}(p, \boldsymbol{x}) = \mathbb{E}_{\boldsymbol{y}\sim q}\left[\rho_b(\boldsymbol{x}, \boldsymbol{y})\right] - \mathbb{E}_{\boldsymbol{x}'\sim p}\left[\rho_b(\boldsymbol{x}, \boldsymbol{x}')\right], \tag{71}$$

and its negative is Eq. (19).

C.6 FD Reward

C.6.1 Proof of Proposition 2

Proposition 2 (FD reward)**.** *Assume the covariance matrices are positive definite and define*

$$G_p = I - \Sigma_q^{1/2} \left(\Sigma_q^{1/2} \Sigma_p \Sigma_q^{1/2}\right)^{-1/2} \Sigma_q^{1/2} \tag{20}$$

for $q \equiv p_{ref}$. A representative of the negative functional derivative of $\mathcal{R}_{\mathrm{FD}}$ is

$$\boxed{r_{\mathrm{FD}}(\boldsymbol{x}; p) = -\left[2(\mu_p - \mu_q)^\top f + (f - \mu_p)^\top G_p (f - \mu_p)\right].} \tag{21}$$

Proof. Write $f = \phi(\boldsymbol{x})$ and recall $\mathcal{R}_{\mathrm{FD}}(p) = \mathrm{FD}(p, p_{\text{ref}})$ from Eq. (5), with μ_p, Σ_p the feature mean and covariance under p. $\mathcal{R}_{\mathrm{FD}}$ depends on p only through its feature mean and covariance. We first differentiate the functional with respect to these two statistics, and then apply the functional chain rule with respect to p.

[5] For experiments in this paper, latent values are bounded and thus satisfy this mild assumption.

Derivatives with respect to the moments. The derivative with respect to the mean is:

$$\frac{\partial \mathcal{R}_{\mathrm{FD}}}{\partial \mu_p} = 2(\mu_p - \mu_q). \tag{72}$$

For the covariance derivative, define

$$S = \Sigma_q^{1/2} \Sigma_p \Sigma_q^{1/2}. \tag{73}$$

Under the positive-definiteness assumption of Proposition 2, the standard matrix differential identity

$$d\ \mathrm{Tr}(S^{1/2}) = \frac{1}{2}\,\mathrm{Tr}\Big(S^{-1/2} dS\Big) \tag{74}$$

gives, since $dS = \Sigma_q^{1/2}(d\Sigma_p)\Sigma_q^{1/2}$,

$$d\ \mathrm{Tr}(S^{1/2}) = \frac{1}{2}\,\mathrm{Tr}\Big(S^{-1/2}\Sigma_q^{1/2}(d\Sigma_p)\Sigma_q^{1/2}\Big) \tag{75}$$

$$= \frac{1}{2}\,\mathrm{Tr}\Big(\Sigma_q^{1/2} S^{-1/2} \Sigma_q^{1/2}\, d\Sigma_p\Big). \tag{76}$$

The covariance-dependent part of Eq. (5) is $\mathrm{Tr}(\Sigma_p) - 2\,\mathrm{Tr}(S^{1/2})$. The first term contributes I, while the calculation above shows that the second contributes $-\Sigma_q^{1/2} S^{-1/2} \Sigma_q^{1/2}$. Hence

$$\frac{\partial \mathcal{R}_{\mathrm{FD}}}{\partial \Sigma_p} = I - \Sigma_q^{1/2}\left(\Sigma_q^{1/2}\Sigma_p\Sigma_q^{1/2}\right)^{-1/2}\Sigma_q^{1/2} =: G_p. \tag{77}$$

Functional derivative with respect to p. Consider a signed perturbation η of p satisfying $\eta(\mathcal{X}) = 0$. The induced first-order change in the mean is

$$d\mu_p[\eta] = \int f(\boldsymbol{x})\,\eta(d\boldsymbol{x}). \tag{78}$$

For the covariance, using $\Sigma_p = \mathbb{E}_p[ff^\top] - \mu_p\mu_p^\top$ gives

$$d\Sigma_p[\eta] = \int f(\boldsymbol{x}) f(\boldsymbol{x})^\top\,\eta(d\boldsymbol{x}) - d\mu_p[\eta]\,\mu_p^\top - \mu_p\, d\mu_p[\eta]^\top \tag{79}$$

$$= \int \left[f(\boldsymbol{x}) f(\boldsymbol{x})^\top - f(\boldsymbol{x})\mu_p^\top - \mu_p f(\boldsymbol{x})^\top\right]\eta(d\boldsymbol{x}). \tag{80}$$

Since $\eta(\mathcal{X}) = 0$, we may add the x-independent matrix $\mu_p\mu_p^\top$ inside the integral without changing its value. Therefore,

$$d\Sigma_p[\eta] = \int (f(\boldsymbol{x}) - \mu_p)(f(\boldsymbol{x}) - \mu_p)^\top\,\eta(d\boldsymbol{x}). \tag{81}$$

Applying the chain rule with Eqs. (72), (77), (78), and (81),

$$d\mathcal{R}_{\mathrm{FD}}(p,q)[\eta] = 2(\mu_p - \mu_q)^\top d\mu_p[\eta] + \langle G_p, d\Sigma_p[\eta]\rangle_F \tag{82}$$

$$= \int \Big[2(\mu_p - \mu_q)^\top f(\boldsymbol{x}) + \big\langle G_p, (f(\boldsymbol{x}) - \mu_p)(f(\boldsymbol{x}) - \mu_p)^\top\big\rangle_F\Big]\eta(d\boldsymbol{x}) \tag{83}$$

$$= \int \Big[2(\mu_p - \mu_q)^\top f(\boldsymbol{x}) + (f(\boldsymbol{x}) - \mu_p)^\top G_p (f(\boldsymbol{x}) - \mu_p)\Big]\eta(d\boldsymbol{x}), \tag{84}$$

where the last equality uses $\langle G, zz^\top\rangle_F = z^\top G z$.

Reading off the integrand, a representative of $\delta\mathcal{R}_{\mathrm{FD}}/\delta p$ is the bracket, and its negative is Eq. (21). □

C.6.2 Connection to Leave-One-Out FD

We next relate the FD reward to a leave-one-out (LOO), or jackknife, construction. Consider the empirical distribution p_N of N samples with features $\{f_i\}_{i=1}^N$, mean

$$\mu = \frac{1}{N}\sum_{i=1}^{N} f_i, \tag{85}$$

and covariance

$$\Sigma = \frac{1}{N}\sum_{i=1}^{N}(f_i-\mu)(f_i-\mu)^\top. \tag{86}$$

Let $p_{N\setminus i}$ denote the empirical distribution obtained after removing sample i, and define the centered feature

$$z_i := f_i - \mu. \tag{87}$$

The leave-one-out mean is

$$\mu_{\setminus i} = \frac{N\mu - f_i}{N-1} = \mu - \frac{z_i}{N-1}, \tag{88}$$

and therefore

$$\mu_{\setminus i} - \mu = -\frac{z_i}{N-1}. \tag{89}$$

The corresponding leave-one-out covariance can be written exactly as

$$\Sigma_{\setminus i} = \frac{N}{N-1}\Sigma - \frac{N}{(N-1)^2} z_i z_i^\top. \tag{90}$$

Hence

$$\Sigma_{\setminus i} - \Sigma = \frac{1}{N-1}\Sigma - \frac{N}{(N-1)^2} z_i z_i^\top. \tag{91}$$

We now expand FD to first order around the moments (μ, Σ) of the full empirical population. Viewing FD as a function of its mean and covariance,

$$\mathcal{R}_{\mathrm{FD}}(p_{N\setminus i}, q) - \mathcal{R}_{\mathrm{FD}}(p_N, q) \approx \left(\frac{\partial \mathcal{R}_{\mathrm{FD}}}{\partial \mu}\right)^\top (\mu_{\setminus i} - \mu) + \left\langle \frac{\partial \mathcal{R}_{\mathrm{FD}}}{\partial \Sigma}, \Sigma_{\setminus i} - \Sigma \right\rangle_F. \tag{92}$$

For the mean contribution, Eqs. (72) and (89) give

$$\left(\frac{\partial \mathcal{R}_{\mathrm{FD}}}{\partial \mu}\right)^\top (\mu_{\setminus i} - \mu) = 2(\mu - \mu_q)^\top \left(-\frac{z_i}{N-1}\right) \tag{93}$$

$$= -\frac{2}{N-1}(\mu - \mu_q)^\top z_i. \tag{94}$$

For the covariance contribution, Eqs. (77) and (91) give

$$\left\langle \frac{\partial \mathcal{R}_{\mathrm{FD}}}{\partial \Sigma}, \Sigma_{\setminus i} - \Sigma \right\rangle_F = \left\langle G, \frac{1}{N-1}\Sigma - \frac{N}{(N-1)^2} z_i z_i^\top \right\rangle_F \tag{95}$$

$$= \frac{1}{N-1}\langle G, \Sigma\rangle_F - \frac{N}{(N-1)^2}\langle G, z_i z_i^\top\rangle_F \tag{96}$$

$$= \frac{1}{N-1}\langle G, \Sigma\rangle_F - \frac{N}{(N-1)^2} z_i^\top G z_i, \tag{97}$$

where we used $\langle G, zz^\top\rangle_F = z^\top G z$.

Combining Eqs. (94) and (97) yields

$$\mathcal{R}_{\mathrm{FD}}(p_{N\setminus i}, q) - \mathcal{R}_{\mathrm{FD}}(p_N, q) \approx -\frac{2}{N-1}(\mu - \mu_q)^\top z_i + \frac{1}{N-1}\langle G, \Sigma\rangle_F - \frac{N}{(N-1)^2} z_i^\top G z_i. \tag{98}$$

Using

$$\frac{N}{(N-1)^2} = \frac{1}{N-1} + O(N^{-2}), \tag{99}$$

the sample-dependent part of Eq. (98) becomes

$$-\frac{1}{N-1}\left[2(\mu-\mu_q)^\top z_i + z_i^\top G z_i\right] + O(N^{-2}). \tag{100}$$

Since $z_i = f_i - \mu$, replacing $2(\mu-\mu_q)^\top z_i$ by $2(\mu-\mu_q)^\top f_i$ changes the expression only by the sample-independent constant $-2(\mu-\mu_q)^\top \mu$. Therefore,

$$\mathcal{R}_{\mathrm{FD}}(p_{N\setminus i}, q) - \mathcal{R}_{\mathrm{FD}}(p_N, q) = \frac{1}{N-1} r_{\mathrm{FD}}(x_i; p_N) + C_N + O(N^{-2}), \tag{101}$$

where C_N is independent of i and

$$r_{\mathrm{FD}}(\boldsymbol{x}_i; p_N) = -\left[2(\mu-\mu_q)^\top f_i + (f_i-\mu)^\top G (f_i-\mu)\right]. \tag{102}$$

Thus, to first order in $1/N$, the FD reward in Proposition 2 agrees with the leave-one-out change in FD up to a common scale and a sample-independent additive constant. The additive constant disappears under reward centering, while the common scale can be absorbed into the reward coefficient. Computationally, this avoids evaluating FD separately for every leave-one-out population: the per-sample rewards share the same population moments and matrix G.

LOO-FID (Baran et al., 2026) computes this leave-one-out difference directly, and differs from Proposition 2 in two respects. First, its reference moments are those of real images rather than p_{ref}. Second, it uses per-dimension variances in place of full covariances, i.e., $\Sigma_p = \mathrm{diag}(\sigma_{p,i}^2), \Sigma_q = \mathrm{diag}(\sigma_{q,i}^2)$. With this specification, G reduces to $\mathrm{diag}(1 - \sigma_{q,i}/\sigma_{p,i})$.

C.6.3 Numerical Stabilization

The main numerical difficulty arises from the covariance component of the FD reward,

$$G_p = I - \Sigma_q^{1/2} S^{-1/2} \Sigma_q^{1/2}, \qquad S = \Sigma_q^{1/2} \Sigma_p \Sigma_q^{1/2}. \tag{103}$$

While the FD value itself contains $S^{1/2}$, its reward requires $S^{-1/2}$, which can strongly amplify errors in small eigenvalues. We therefore regularize the covariance estimates using shrinkage.

Covariance shrinkage. Small covariance eigenvalues are stabilized by applying the same shrinkage map to the generated and reference covariances:

$$\widetilde{\Sigma} = (1-\nu)\Sigma + \nu\,\tau I, \qquad \tau = \frac{\mathrm{Tr}(\Sigma)}{d}, \qquad 0 \le \nu < 1. \tag{104}$$

Shrinkage suppresses noisy low-variance directions while preserving the matching fixed point: if $\Sigma_p = \Sigma_q$, then $\widetilde{\Sigma}_p = \widetilde{\Sigma}_q$.

Let

$$\widetilde{G}_p = I - \widetilde{\Sigma}_q^{1/2}\left(\widetilde{\Sigma}_q^{1/2}\widetilde{\Sigma}_p\widetilde{\Sigma}_q^{1/2}\right)^{-1/2}\widetilde{\Sigma}_q^{1/2}. \tag{105}$$

Because the shrinkage target $\tau = \mathrm{Tr}(\Sigma)/d$ itself depends on the covariance, simply substituting $\widetilde{G}_p$ for G_p does not give the exact gradient of FD between the shrunk covariances. From

$$d\widetilde{\Sigma}_p = (1-\nu)d\Sigma_p + \frac{\nu}{d}\,\mathrm{Tr}(d\Sigma_p) I, \tag{106}$$

the corresponding corrected matrix is

$$G_p^{\text{eff}} = (1-\nu)\widetilde{G}_p + \frac{\nu}{d}\,\mathrm{Tr}(\widetilde{G}_p) I. \tag{107}$$

We therefore replace G_p by G_p^{eff} in the covariance term of the reward. This retains the exact score-function gradient of the FD objective computed using the shrunk covariances. At the matching fixed point, $\widetilde{G}_p = G_p^{\text{eff}} = 0$.

Moment estimation and diagnostics. Following Yang et al. (2026), we estimate the generated moments using an exponential moving average of the raw first and second moments,

$$\mu_p \leftarrow \kappa\mu_p + (1-\kappa)\mu_{\text{batch}}, \qquad M_p \leftarrow \kappa M_p + (1-\kappa)M_{\text{batch}}, \tag{108}$$

and form $\Sigma_p = M_p - \mu_p\mu_p^\top$. The EMA decouples the population used to estimate the covariance from the current minibatch size, providing a substantially larger effective sample population for the FD statistics. All covariance construction and eigendecomposition operations are performed in double precision.

Calibration of the FD coefficient. We build the reference statistics once from 25×1152 samples of p_{ref} at the training sampling settings. For each feature space i, we divide the FD by its noise floor D_i, the FD between two disjoint halves of the reference samples, so that each space is measured in units of its own finite-sample noise. In a run that logs the FD reward without optimizing it, we measure each space's reward spread relative to the advantage spread, s_i, after the moment EMA has turned over (training steps 20–30). We weight the spaces by $w_i = \bar{s}/s_i$ so that they contribute equally, and set $\lambda = \phi/s_w$, where s_w is the spread of the weighted combined reward, accounting for the correlation between spaces. The FD reward then contributes a fraction ϕ of the advantage spread at the calibration point; once training reduces the FD below that point, the realized fraction is smaller. The moment EMA uses $\kappa = 0.95$ and covariance shrinkage $\nu = 0.02$.

C.7 DreamSim diversity Reward

Proposition 3 (DreamSim reward)**.** *A representative of the negative functional derivative of $\mathcal{R}_{\text{DS}}$ is*

$$\boxed{r_{\text{DS}}(\boldsymbol{x}; p) = -z(\boldsymbol{x})^\top \mathbb{E}_{\boldsymbol{x}'\sim p}[z(\boldsymbol{x}')].} \tag{22}$$

Proof. Assume $\phi_{\text{DS}}(\boldsymbol{x}) \neq 0$ for every $\boldsymbol{x}$, so that $z(\boldsymbol{x})$ is defined, and write

$$s(\boldsymbol{x}, \boldsymbol{x}') = z(\boldsymbol{x})^\top z(\boldsymbol{x}') \tag{109}$$

for the cosine similarity. Since $\|z(\boldsymbol{x})\|_2 = 1$, $|s(\boldsymbol{x}, \boldsymbol{x}')| \leq 1$, so every integral of s against finite (signed) measures below is finite; unlike Appendix C.5, no moment assumption on p is needed.

By Eq. (7), $1 - \text{Div}_{\text{DS}}(p) = \mathbb{E}_{\boldsymbol{x},\boldsymbol{x}'\sim p}[s(\boldsymbol{x}, \boldsymbol{x}')]$, so

$$\mathcal{R}_{\text{DS}}(p) = \frac{1}{2}\iint s(\boldsymbol{x}, \boldsymbol{x}')\, p(\mathrm{d}\boldsymbol{x})\, p(\mathrm{d}\boldsymbol{x}'). \tag{110}$$

This has the form of the term T_2 in Appendix C.5, with ρ_b replaced by s and an overall factor $\frac{1}{2}$.

Let η be a signed measure with $\eta(\mathcal{X}) = 0$ and finite total variation $|\eta|(\mathcal{X})$, and replace p by $p + \varepsilon\eta$. Expanding the product $(p + \varepsilon\eta)(p + \varepsilon\eta)$ in Eq. (110),

$$\begin{aligned}\mathcal{R}_{\text{DS}}(p + \varepsilon\eta) = \mathcal{R}_{\text{DS}}(p) &+ \frac{\varepsilon}{2}\iint s(\boldsymbol{x}, \boldsymbol{x}')\, \eta(\mathrm{d}\boldsymbol{x})\, p(\mathrm{d}\boldsymbol{x}') + \frac{\varepsilon}{2}\iint s(\boldsymbol{x}, \boldsymbol{x}')\, p(\mathrm{d}\boldsymbol{x})\, \eta(\mathrm{d}\boldsymbol{x}') \\ &+ \frac{\varepsilon^2}{2}\iint s(\boldsymbol{x}, \boldsymbol{x}')\, \eta(\mathrm{d}\boldsymbol{x})\, \eta(\mathrm{d}\boldsymbol{x}').\end{aligned} \tag{111}$$

The two first-order terms are equal: renaming the integration variables $\boldsymbol{x} \leftrightarrow \boldsymbol{x}'$ in the second and using the symmetry $s(\boldsymbol{x}, \boldsymbol{x}') = s(\boldsymbol{x}', \boldsymbol{x})$ turns it into the first.

$$\mathcal{R}_{\text{DS}}(p+\varepsilon\eta) = \mathcal{R}_{\text{DS}}(p) + \varepsilon\int\Big[\int s(\boldsymbol{x}, \boldsymbol{x}')\, p(\mathrm{d}\boldsymbol{x}')\Big]\eta(\mathrm{d}\boldsymbol{x}) + \frac{\varepsilon^2}{2}\iint s(\boldsymbol{x}, \boldsymbol{x}')\, \eta(\mathrm{d}\boldsymbol{x})\, \eta(\mathrm{d}\boldsymbol{x}'). \tag{112}$$

Since $|s| \leq 1$, its ε^2 coefficient is bounded in absolute value by $\frac{1}{2}\,|\eta|(\mathcal{X})^2$. Therefore,

$$\lim_{\varepsilon\to 0}\frac{\mathcal{R}_{\text{DS}}(p + \varepsilon\eta) - \mathcal{R}_{\text{DS}}(p)}{\varepsilon} = \int \mathbb{E}_{\boldsymbol{x}'\sim p}\big[s(\boldsymbol{x}, \boldsymbol{x}')\big]\ \eta(\mathrm{d}\boldsymbol{x}). \tag{113}$$

Because $z(\boldsymbol{x})$ does not depend on $\boldsymbol{x}'$, it can be taken out of the inner expectation:

$$\mathbb{E}_{\boldsymbol{x}'\sim p}\big[s(\boldsymbol{x}, \boldsymbol{x}')\big] = \mathbb{E}_{\boldsymbol{x}'\sim p}\big[z(\boldsymbol{x})^\top z(\boldsymbol{x}')\big] = z(\boldsymbol{x})^\top \mathbb{E}_{\boldsymbol{x}'\sim p}[z(\boldsymbol{x}')]. \tag{114}$$

Comparing with Eq. (9), a representative of the functional derivative is

$$\frac{\delta \mathcal{R}_{\mathrm{DS}}}{\delta p}(p, \boldsymbol{x}) = z(\boldsymbol{x})^\top \mathbb{E}_{\boldsymbol{x}' \sim p}[z(\boldsymbol{x}')], \tag{115}$$

and its negative is Eq. (22). □

Group estimate. For a group $\{\boldsymbol{x}^i\}_{i=1}^G$ of i.i.d. samples from p, the population mean $m(p)$ is estimated from the other samples in the group, excluding the sample itself (Section 3.2):

$$\hat{r}_{\mathrm{DS}}(\boldsymbol{x}^i) = -\frac{1}{G-1} \sum_{j \neq i} z(\boldsymbol{x}^i)^\top z(\boldsymbol{x}^j). \tag{116}$$

Conditionally on $\boldsymbol{x}^i$, each $\boldsymbol{x}^j$ with $j \neq i$ is an independent draw from p, so

$$\mathbb{E}\big[\hat{r}_{\mathrm{DS}}(\boldsymbol{x}^i) \,\big|\, \boldsymbol{x}^i\big] = -z(\boldsymbol{x}^i)^\top m(p) = r_{\mathrm{DS}}(\boldsymbol{x}^i; p) \tag{117}$$

exactly; including $j = i$ would instead add the sample-independent term $-\|z(\boldsymbol{x}^i)\|_2^2/G = -1/G$ and rescale by $(G-1)/G$. Moreover, adding and subtracting 1 inside the sum,

$$\hat{r}_{\mathrm{DS}}(\boldsymbol{x}^i) = \frac{1}{G-1} \sum_{j \neq i} \Big[1 - z(\boldsymbol{x}^i)^\top z(\boldsymbol{x}^j)\Big] - 1, \tag{118}$$

so up to the constant -1, which group centering removes, the estimate is the mean DreamSim cosine distance from $\boldsymbol{x}^i$ to the other samples in its group.

DRIFT (Liu et al., 2026, Section 5.3 and Appendix C) rewards each sample with the average squared embedding distance to the other samples in its group, $d(\boldsymbol{x}^i) = \frac{1}{G-1} \sum_{j \neq i} \|E(\boldsymbol{x}^i) - E(\boldsymbol{x}^j)\|_2^2$. With the normalized DreamSim feature $E = z$, $\|z(\boldsymbol{x}^i) - z(\boldsymbol{x}^j)\|_2^2 = 2 - 2\, z(\boldsymbol{x}^i)^\top z(\boldsymbol{x}^j)$, so $d(\boldsymbol{x}^i) = 2\,\hat{r}_{\mathrm{DS}}(\boldsymbol{x}^i) + 2$ for every sample: after group centering, DRIFT's reward is $2\hat{r}_{\mathrm{DS}}$, and the factor 2 is absorbed into λ. DRIFT additionally clips this reward and standardizes its advantage separately.

D Experiment Details

D.1 Evaluation metrics

Drift. We measure drift against p_{ref} rather than against a fixed real dataset, which separates the change induced by fine-tuning from the pretrained model's own mismatch with real data. Computing FD per prompt and then averaging measures this change within each conditional distribution, rather than in the marginal over prompts. Its absolute values are far larger than standard FD values and are meaningful only relative to one another. We compute avg(FD-64) with Inception-v3 and DINOv2 ViT-L/14, and DreamSim diversity with DINO ViT-B/16 and OpenCLIP ViT-B/32.

Held-out metrics. DFR-FD computes its reward with ConvNeXt-V2, SigLIP2, and CLIP ViT-L/14 features, and DFR-DS with the DINO ViT-B/16 branch of DreamSim. A metric is not held out for an instance if it is computed with one of that instance's encoders or a closely related one, because the instance optimizes a closely related quantity during training. We therefore strike through DreamSim's DINO branch and DINOv2 FD, a later model of the same self-supervised family, for DFR-DS. DreamSim's OpenCLIP branch remains held out for DFR-FD: it is a ViT-B/32 trained on different data and fine-tuned for perceptual similarity, not the CLIP ViT-L/14 that DFR-FD uses. Struckout entries are not counted as wins or losses. DFR-MMD compares SD3.5-M's own VAE latents and uses no pretrained feature encoder, so every metric is held out for it.

D.2 Evaluation protocol

Checkpoint selection. During training, we evaluate the target reward every 10 steps on the task's full test prompt set. This curve is used to select checkpoints: when all three seeds possess a checkpoint that reaches the threshold within the compute budget, it is considered successful. The full tables (Appendix D.6) also report configurations that do not reach a threshold: seeds that reach it are evaluated at their first checkpoint that reaches it and the others at their peak, taking the earliest

on ties. Steps is the mean number of training steps to the threshold over the seeds that reach it, from which Appendix D.7 derives the speedups.

Evaluation prompts and sampling. From the GenEval test set ("g"; 2,212 prompts), DrawBench ("d"; 1,000), and the OCR test set ("o"; 1,018), we draw a fixed random subset of 256 prompts each. Models fine-tuned on GenEval are evaluated on g and d, and models fine-tuned on OCR on o and d. For every prompt, every model generates 64 images from the same 64 initial-noise seeds, at 512×512 resolution with 20 sampling steps and guidance scale 4.5; multi-reward DiffusionNFT is sampled at its prescribed scale of 1.0. We run the transformer and text encoders in FP16 and decode with the VAE in FP32.

Statistics. RAM and DFR entries are the mean $\pm$ sample SD over three seeds; existing methods are single checkpoints and are reported without SD. Two configurations are tied on a metric if their means, as displayed, differ by at most the larger of their two SDs, and otherwise the one with the better mean wins. In the ImgRwd and PickScore columns of the full tables (Appendix D.6), red marks a value significantly below SD3.5-M's by the same criterion; for single-seed methods, we use the median seed SD of the RAM rows.

D.3 Compute budget

All runs of a task use the same compute: a fixed wall-clock budget of 288 GPU hours for GenEval and 144 for OCR. The distributional rewards add little time per training step. Measured on this hardware, DFR-FD and DFR-DS are within 3% of RAM, and DFR-MMD takes approximately 9% longer, mostly for the pairwise distances within each group and to the reference samples. Within the same budget, DFR-MMD therefore completes correspondingly fewer steps. Speedups in Table 1 include this overhead.

D.4 Hyperparameters

Table 4: Training hyperparameters, shared by RAM and every DFR instance on both tasks. Only β and the DFR terms differ between configurations.

Hyperparameter	**Value**
Base model	SD3.5-M
Fine-tuning	LoRA, rank 32, scaling 64
Optimizer	AdamW
Learning rate	3×10^{-4}
Adam (β_1, β_2)	$(0.9,\ 0.99)$
Weight decay	10^{-4}
Precision	FP16 mixed precision
Prompts per training step	48
Group size G (samples per prompt)	24
Optimizer steps per training step	1
Sampling during training	20 steps, guidance scale 2.0, 512×512
Loss timesteps per sample	8, sampled with weight decreasing linearly toward low noise
Advantage	group-centered, divided by the reward SD over the training step
Advantage scale $1/\alpha$	100
Lagged policy v_{old}	EMA of the trained weights, decay 0.9
Reverse-KL coefficient β	swept (Section 4.1)
Validation	every 10 steps, guidance scale 4.5

DFR coefficients. Each instance adds its group-centered reward $r_{\mathcal{R}}$ to the normalized advantage with coefficient λ (Algorithm 1), so λ is in units of the task advantage. DFR-MMD uses the energy kernel with exponent $b = 1.75$ on SD3.5-M's normalized VAE latents, with $M = 8$ reference samples per prompt from p_{ref}. DFR-DS uses the DINO branch of DreamSim with distances normalized to $[0, 1]$. DFR-FD combines three feature spaces, each measured in units of its own noise floor, so

its raw scale is not interpretable; we instead set $\lambda = \phi / s_w$, where ϕ is the fraction of the advantage spread that the FD reward contributes and s_w is calibrated once per task (Appendix C.6.3). The tables use $\lambda = 0.55$ (MMD), $\phi = 0.07$ (FD), and $\lambda = 3$ (DS) on GenEval, and $\lambda = 1.5$, $\phi = 0.09$, and $\lambda = 4$ on OCR.

Choice of β. We add DFR at the largest β for which RAM reaches the highest threshold on every seed. DFR-DS keeps this β: its diversity reward does not measurably slow reward optimization, so it adds diversity at no cost in speed rather than acting as further regularization. DFR-MMD and DFR-FD also pull the policy toward p_{ref}; on OCR they no longer reach the threshold on every seed at $\beta = 5$, so we use $\beta = 4$.

D.5 Baselines

We compare with ELBO-based DiffusionNFT and AWM, and with trajectory-based Flow-GRPO and zeroth-order RSM. For Flow-GRPO, we use the original released GenEval checkpoint (revision `5efdfea`). For multi-reward DiffusionNFT, we use the released checkpoint, and sample at its prescribed guidance scale of 1.0. We reproduce GenEval-only DiffusionNFT, AWM, and RSM with one seed each, using their released code and default settings, with the exception that we adjust AWM's effective batch size configuration to match other methods. We evaluate each reproduced run at its first checkpoint whose test reward reaches 0.95; RSM's reward jumps from 0.94 to 0.97 between consecutive checkpoints, so we evaluate it at 0.94. Their speedups are those reported by the respective papers (Appendix D.7).

D.6 Full Table

Tables 5 and 6 extend Tables 1 and 2 with the DINO branch of DreamSim, held-out preference rewards, and the number of training steps to each threshold. They also include the configurations that do not reach a threshold on every seed, evaluated at the first checkpoint that reaches it or, failing that, at their peak (Appendix D.2); these rows are evaluated at a lower reward and are therefore not used to color the DFR entries.

DreamSim branches. The DINO branch agrees with the OpenCLIP branch reported in the main text: on GenEval, DFR-MMD and DFR-FD win against every RAM configuration reaching the same threshold on the test prompts at both thresholds, and on OCR they tie or win.

Preference rewards. No DFR instance scores significantly below SD3.5-M on GenEval, whereas several existing methods do on DrawBench. On OCR, RAM ($\beta = 2$) falls significantly below SD3.5-M on both preference rewards and both prompt sets, consistent with the instability of small β (Appendix D.9); DFR-DS and RAM ($\beta = 5$) are marginally below it in PickScore on DrawBench.

Multi-reward training. Multi-reward DiffusionNFT, trained on PickScore and related preference rewards in addition to GenEval, raises both preference rewards well above those of every other method, yet drifts more than GenEval-only DiffusionNFT in DINOv2 features ($1468/1702$ versus $1352/1437$) and is no more diverse ($0.102/0.106$ versus $0.109/0.143$). Training on more rewards is therefore not in itself a remedy for distribution drift or mode collapse.

D.7 Speedups

Table 7 lists the speedups over Flow-GRPO reported by existing methods and derives ours: we measure the time to reach the threshold relative to RAM ($\beta = 1$) on GenEval and RAM ($\beta = 2$) on OCR, as training steps times the time per step on the same hardware (Appendix D.3), and scale RAM's reported speedup accordingly.

D.8 Held-out rewards

Figure 3 evaluates checkpoints of RAM ($\beta = 1$) every 30 steps with the protocol of Appendix D.2. Within the first 30 steps, ImageReward rises from 0.84 to 1.12 on the GenEval prompts and from 0.86 to 1.03 on DrawBench, while DreamSim diversity falls from 0.302 to 0.179 and from 0.274 to 0.186. Over the rest of training, ImageReward keeps rising on the GenEval prompts, to 1.19–1.23 from step 120 on, and stays between 0.98 and 1.09 on DrawBench, while diversity stays far below

Table 5: All metrics for models fine-tuned on GenEval, at thresholds 0.95 and 0.96. Reward is the threshold for configurations that reach it and, otherwise, the mean GenEval at the evaluated checkpoints; Steps is the mean number of training steps to the threshold. Entries of the form a/b report results on the GenEval test prompts (a) and on DrawBench (b), as means over three seeds with SD beneath; existing methods are single checkpoints. Greyed-out rows (†) do not reach the threshold on every seed. Blue and struck-through entries are as in Table 1; in the ImgRwd (ImageReward) and PickScore columns, red marks a value significantly below that of SD3.5-M (Appendix D.2).

Method	Reward	Steps (↓)	FID (↓)	FD-DINOv2 (↓)	DreamSim-DINO (↑)	DreamSim-OpenCLIP (↑)	ImgRwd (↑)	PickScore (↑)
GenEval, threshold 0.95								
SD3.5-M	0.63	–	–	–	0.405/0.372	0.302/0.274	0.840/0.859	0.872/0.863
Flow-GRPO	0.95	–	114.32/125.61	1019/997	0.224/0.274	0.155/0.189	1.222/1.013	0.883/0.862
RSM[§]	0.94	–	132.82/154.29	1271/1310	0.184/0.249	0.142/0.183	1.082/0.838	0.869/0.851
AWM*	0.95	–	158.88/183.71	1412/1578	0.124/0.173	0.095/0.132	1.028/0.573	0.869/0.840
DiffusionNFT	0.96	–	144.36/159.22	1352/1437	0.151/0.204	0.109/0.143	1.232/1.091	0.876/0.851
DiffusionNFT (multi)[‡]	0.94	–	145.06/175.20	1468/1702	0.143/0.141	0.102/0.106	~~1.481/1.457~~	~~0.931/0.915~~
RAM ($\beta = 1$)	0.95	157 ±55	132.93/144.77 ±16.19/16.96	1211/1236 ±116/185	0.225/0.276 ±0.037/0.027	0.163/0.193 ±0.021/0.021	1.151/1.013 ±0.109/0.152	0.872/0.857 ±0.010/0.010
RAM ($\beta = 2$)	0.95	143 ±15	116.99/122.78 ±1.69/1.02	1078/1002 ±34/11	0.239/0.301 ±0.019/0.012	0.170/0.208 ±0.009/0.006	1.209/1.069 ±0.016/0.014	0.880/0.865 ±0.002/0.001
RAM ($\beta = 3$)	0.95	187 ±21	113.19/122.06 ±2.83/1.84	1030/991 ±11/9	0.260/0.310 ±0.009/0.011	0.184/0.214 ±0.006/0.010	1.226/1.082 ±0.020/0.009	0.884/0.866 ±0.002/0.001
RAM ($\beta = 4$)[†]	0.94	210 ±28	115.49/126.77 ±7.07/14.99	1059/1053 ±107/168	0.249/0.305 ±0.030/0.023	0.181/0.220 ±0.015/0.012	1.204/0.964 ±0.044/0.157	0.884/0.862 ±0.003/0.007
DFR-MMD ($\beta = 2$, $\lambda = 0.55$)	0.95	183 ±40	111.20/120.64 ±4.10/6.17	990/967 ±27/52	0.295/0.326 ±0.025/0.014	0.200/0.224 ±0.015/0.011	1.200/1.049 ±0.042/0.043	0.882/0.863 ±0.005/0.003
DFR-FD ($\beta = 2$)	0.95	223 ±29	111.02/117.09 ±2.00/4.88	980/932 ±20/46	0.282/0.332 ±0.005/0.007	0.199/0.231 ±0.003/0.006	1.199/1.034 ±0.013/0.048	0.882/0.866 ±0.003/0.002
DFR-DS ($\beta = 2$)	0.95	197 ±25	114.60/121.86 ±1.64/4.22	~~1034/992~~ ~~±37/82~~	~~0.325/0.344~~ ~~±0.017/0.012~~	0.218/0.233 ±0.007/0.010	1.185/1.032 ±0.018/0.041	0.881/0.863 ±0.006/0.005
GenEval, threshold 0.96								
SD3.5-M	0.63	–	–	–	0.405/0.372	0.302/0.274	0.840/0.859	0.872/0.863
RAM ($\beta = 1$)[†]	0.95	195 ±49	127.57/137.62 ±8.73/10.49	1171/1166 ±86/108	0.229/0.277 ±0.012/0.016	0.161/0.190 ±0.007/0.008	1.208/1.057 ±0.053/0.056	0.875/0.859 ±0.008/0.005
RAM ($\beta = 2$)	0.96	210 ±20	119.87/125.99 ±2.03/3.89	1066/1023 ±31/33	0.249/0.310 ±0.004/0.005	0.174/0.213 ±0.006/0.006	1.202/1.046 ±0.035/0.060	0.884/0.865 ±0.003/0.002
RAM ($\beta = 3$)[†]	0.95	220	113.91/121.93 ±6.49/5.44	1041/993 ±49/53	0.254/0.308 ±0.006/0.006	0.181/0.214 ±0.005/0.008	1.231/1.087 ±0.019/0.008	0.885/0.866 ±0.004/0.003
RAM ($\beta = 4$)[†]	0.95	–	116.29/129.16 ±6.13/13.02	1065/1073 ±102/156	0.247/0.305 ±0.030/0.023	0.180/0.221 ±0.015/0.011	1.201/0.955 ±0.037/0.147	0.884/0.862 ±0.004/0.007
DFR-MMD ($\beta = 2$)	0.96	223 ±47	110.75/121.67 ±3.04/6.05	992/977 ±5/47	0.290/0.321 ±0.021/0.015	0.196/0.221 ±0.011/0.009	1.209/1.030 ±0.016/0.025	0.884/0.864 ±0.002/0.001
DFR-FD ($\beta = 2$)	0.96	260 ±26	111.95/120.40 ±4.11/4.05	994/969 ±46/38	0.273/0.319 ±0.021/0.018	0.193/0.223 ±0.011/0.013	1.209/1.027 ±0.005/0.032	0.883/0.865 ±0.004/0.003
DFR-DS ($\beta = 2$)	0.96	240 ±44	114.47/124.11 ±0.31/3.88	~~1052/1007~~ ~~±8/45~~	~~0.332/0.349~~ ~~±0.028/0.019~~	0.225/0.240 ±0.022/0.018	1.170/0.984 ±0.046/0.082	0.878/0.860 ±0.004/0.004

[†]Not every seed reaches the threshold; Steps averages the seeds that do.
[§], *, [‡] as in Table 1; multi-reward DiffusionNFT is trained on PickScore and closely related preference rewards, so its held-out rewards are struck through.

Table 6: All metrics for models fine-tuned on OCR, at threshold 0.96. Columns and markings are as in Table 5, except that entries report results on the OCR test prompts (a) and on DrawBench (b).

Method	Reward	Steps (↓)	FID (↓)	FD-DINOv2 (↓)	DreamSim-DINO (↑)	DreamSim-OpenCLIP (↑)	ImgRwd (↑)	PickScore (↑)
OCR, threshold 0.96								
SD3.5-M	0.59	–	–	–	0.307/0.372	0.258/0.274	1.004/0.859	0.863/0.863
RAM ($\beta = 2$)	0.96	67 ±12	219.07/131.38 ±24.04/7.71	1270/1040 ±232/84	0.248/0.347 ±0.016/0.010	0.128/0.245 ±0.010/0.009	0.545/0.712 ±0.251/0.133	0.849/0.855 ±0.009/0.004
RAM ($\beta = 3$)[†]	0.95	100 ±0	149.72/104.84 ±5.66/3.13	728/798 ±40/58	0.272/0.349 ±0.019/0.011	0.167/0.246 ±0.007/0.004	1.039/0.894 ±0.054/0.048	0.869/0.862 ±0.003/0.002
RAM ($\beta = 4$)	0.96	123 ±15	138.04/100.94 ±6.73/4.36	667/751 ±46/56	0.291/0.355 ±0.010/0.008	0.185/0.249 ±0.004/0.005	1.062/0.922 ±0.064/0.045	0.867/0.862 ±0.003/0.002
RAM ($\beta = 5$)	0.96	137 ±6	136.78/97.16 ±5.35/2.14	662/721 ±39/21	0.292/0.358 ±0.003/0.004	0.192/0.252 ±0.005/0.003	1.040/0.886 ±0.043/0.030	0.867/0.861 ±0.002/0.001
RAM ($\beta = 6$)[†]	0.95	180 ±28	121.69/93.58 ±5.38/2.03	565/695 ±31/26	0.285/0.345 ±0.006/0.011	0.192/0.244 ±0.003/0.010	1.100/0.928 ±0.031/0.023	0.871/0.863 ±0.002/0.002
DFR-MMD ($\beta = 4$)	0.96	153 ±12	124.87/95.37 ±0.83/1.60	597/708 ±5/16	0.300/0.367 ±0.006/0.012	0.197/0.261 ±0.006/0.008	1.057/0.862 ±0.005/0.017	0.868/0.861 ±0.001/0.001
DFR-FD ($\beta = 4$)	0.96	183 ±23	121.46/94.56 ±4.52/1.20	585/705 ±20/13	0.295/0.359 ±0.009/0.010	0.213/0.255 ±0.004/0.008	1.097/0.886 ±0.062/0.057	0.870/0.861 ±0.004/0.004
DFR-DS ($\beta = 5$)	0.96	143 ±15	136.13/99.23 ±4.59/3.20	~~653/743~~ ~~±37/27~~	~~0.355/0.399~~ ~~±0.010/0.004~~	0.224/0.281 ±0.008/0.003	1.013/0.837 ±0.035/0.024	0.864/0.858 ±0.002/0.001

[†]Not every seed reaches the threshold; Steps averages the seeds that do.

SD3.5-M's, at 0.155–0.168 and 0.174–0.201 from step 60 on. ImageReward thus rates a model higher than SD3.5-M even after its diversity on the GenEval prompts has roughly halved: it scores each image in isolation and does not register that the images for a prompt have become alike. We therefore report held-out rewards (Appendix D.6) but do not use them to measure reward hacking.

D.9 Training stability across β

Training is unstable at both ends of the range of β, in different ways (Figure 4). At $\beta = 1$, runs disagree: the seed SDs of FD at the evaluated checkpoints are 3–10× those at $\beta \in \{2, 3\}$ (Table 5), and the reward dips by up to 0.037 below its running maximum, but every dip recovers and no seed collapses within the budget. On OCR, the smallest $\beta = 2$ likewise has FD seed SDs 4–6× those at

Table 7: Speedups. Top: speedups over Flow-GRPO reported by the respective papers. Bottom: measured from our runs. Time is the number of steps to the threshold multiplied by the time per step, both relative to the RAM configuration stated in each block. GenEval speedups divide the reported speedup of RAM ($\beta = 1$) by Time; OCR has no Flow-GRPO reference and reports Time only. Means with SD beneath are over seeds; "–" marks a threshold not reached on every seed.

Method	**Speedup**	**Source**
Flow-GRPO	1	reference
RSM	6.25	$5\times$ over TempFlow-GRPO (Lee et al., 2026), which reports $1.25\times$ over Flow-GRPO (He et al., 2026)
AWM	8	reported by Xue et al. (2026)
DiffusionNFT	24	reported by Zheng et al. (2026)
DiffusionNFT (multi)	2.28	reported by Choi et al. (2026)
RAM ($\beta = 1$)	52.3	$50\times$ fewer steps, times the ratio of time per step (Bergmeister et al., 2026)

Method	**Steps (per seed)**	**Time per step**	**Time** ($\downarrow$)	**Speedup** ($\uparrow$)
GenEval, threshold 0.95*; Time relative to RAM (*$\beta = 1$*)*				
RAM ($\beta = 1$)	160/100/210	1.00	1.00	52.3
RAM ($\beta = 2$)	160/140/130	1.00	0.91 $\pm$0.10	57.2 $\pm$6.1
RAM ($\beta = 3$)	210/180/170	1.00	1.19 $\pm$0.13	43.9 $\pm$4.9
RAM ($\beta = 4$)	–	1.00	–	–
DFR-MMD ($\beta = 2$)	230/160/160	1.09	1.27 $\pm$0.28	41.1 $\pm$9.1
DFR-FD ($\beta = 2$)	240/240/190	1.03	1.47 $\pm$0.19	35.7 $\pm$4.6
DFR-DS ($\beta = 2$)	200/170/220	1.03	1.29 $\pm$0.17	40.4 $\pm$5.2
OCR, threshold 0.96*; Time relative to RAM (*$\beta = 2$*)*				
RAM ($\beta = 2$)	60/80/60	1.00	1.00	–
RAM ($\beta = 3$)	–	1.00	–	–
RAM ($\beta = 4$)	110/120/140	1.00	1.85 $\pm$0.23	–
RAM ($\beta = 5$)	140/130/140	1.00	2.05 $\pm$0.09	–
RAM ($\beta = 6$)	–	1.00	–	–
DFR-MMD ($\beta = 4$)	160/140/160	1.09	2.50 $\pm$0.19	–
DFR-FD ($\beta = 4$)	170/210/170	0.98	2.70 $\pm$0.34	–
DFR-DS ($\beta = 5$)	160/130/140	0.97	2.08 $\pm$0.22	–

$\beta \in \{4, 5\}$ (Table 6). At $\beta = 4$ of GenEval, training collapses: every seed peaks between steps 170 and 240 and then declines. On OCR, larger β plateaus rather than collapses: RAM peaks at 0.950 at $\beta = 8$ and 0.923 at $\beta = 12$.

E Additional Results

E.1 Qualitative comparison between generated images

We demonstrate that DFR produces a distribution that more closely matches that of the base SD3.5 model (see below). For the donut example, DFR preserves the color of the glaze better than the RAM baseline. For the four-clock example, DFR captures both the clock colors and the mix of Roman and Arabic numerals, whereas the RAM baseline collapses these variations. These examples qualitatively highlight the strengths of DFR.

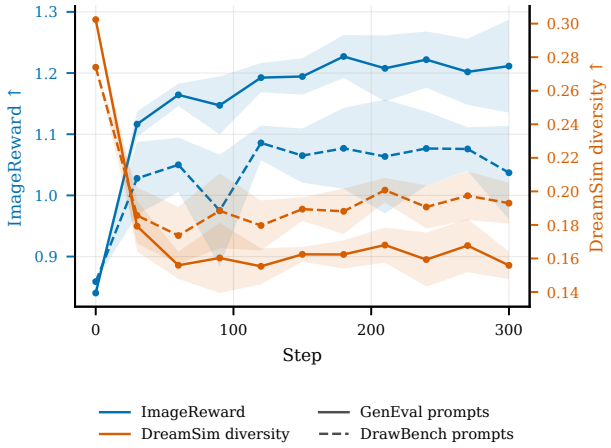
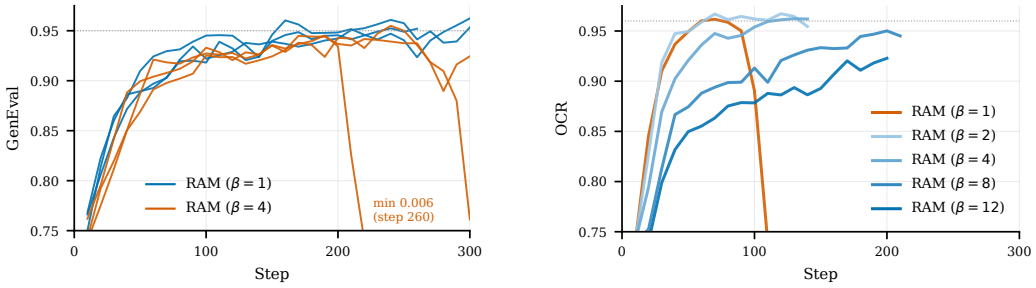


Figure 3: ImageReward (left axis) and DreamSim diversity (right axis) of RAM ($\beta = 1$) during training on GenEval, on the GenEval (solid) and DrawBench (dashed) prompts, evaluated every 30 steps; step 0 is SD3.5-M. Mean over three seeds with ± 1 SD bands.



(a) GenEval. (b) OCR.

Figure 4: Target reward on held-out prompts during RAM training. **Left:** GenEval at $\beta = 1$ and $\beta = 4$, with one line per seed and the target indicated at 0.95. At $\beta = 1$, no seed collapses within the training budget; at $\beta = 4$, every seed declines after peaking, and one collapses. **Right:** OCR for $\beta \in \{1, 2, 4, 8, 12\}$, using 1 seed for each coefficient and a target of 0.96. The weakly regularized $\beta = 1$ run collapses after reaching its peak, whereas the more strongly regularized runs remain stable over their observed training ranges, at the cost of slower reward improvement for $\beta = 8$ and $\beta = 12$.

E.2 Samples from baselines

We include samples generated from the base SD3.5 model, RAM, and DFR below.

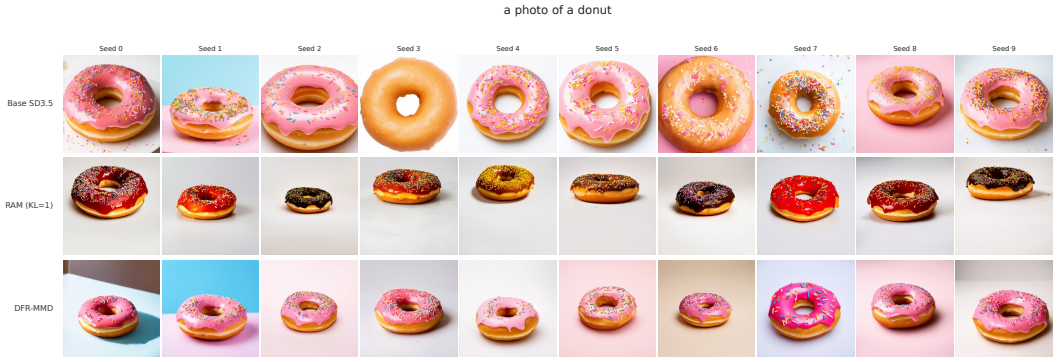


Figure 5: DFR qualitatively appears closer to the pretrained distribution; the colour of the glazing on the donut is preserved in DFR, while this does not hold for the RAM baseline.

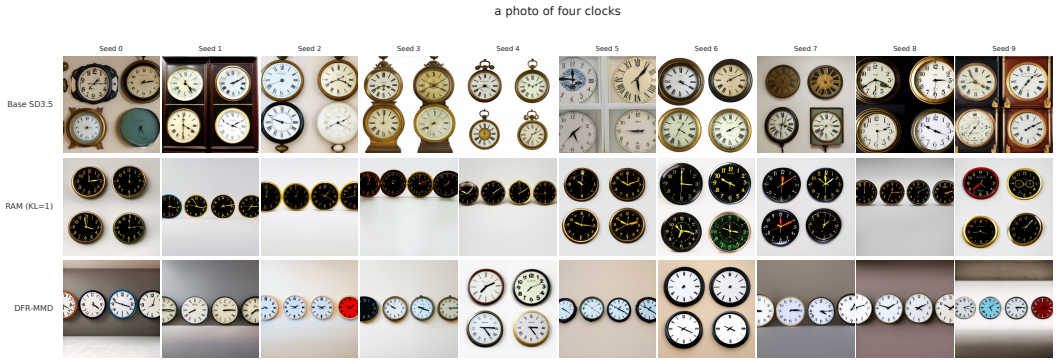


Figure 6: DFR captures both the clock colors and the mix of Roman and Arabic numerals, preserving variations from the pretrained SD3.5 distribution, whereas the RAM baseline collapses these variations.



Figure 7: Sample images generated for diverse prompts across various seeds using the base SD3.5 model.

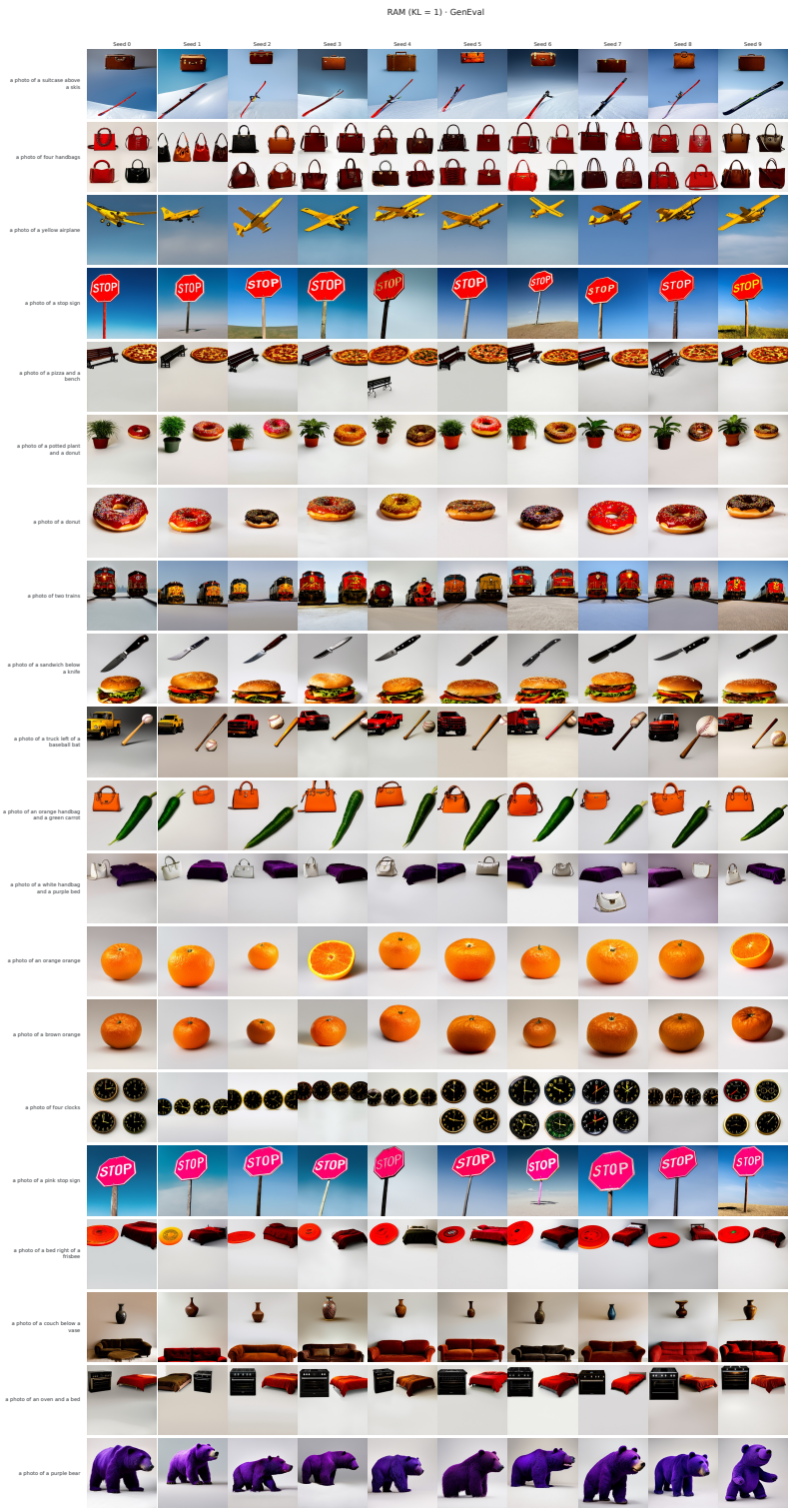


Figure 8: Sample images generated for diverse prompts across various seeds using the RAM baseline.

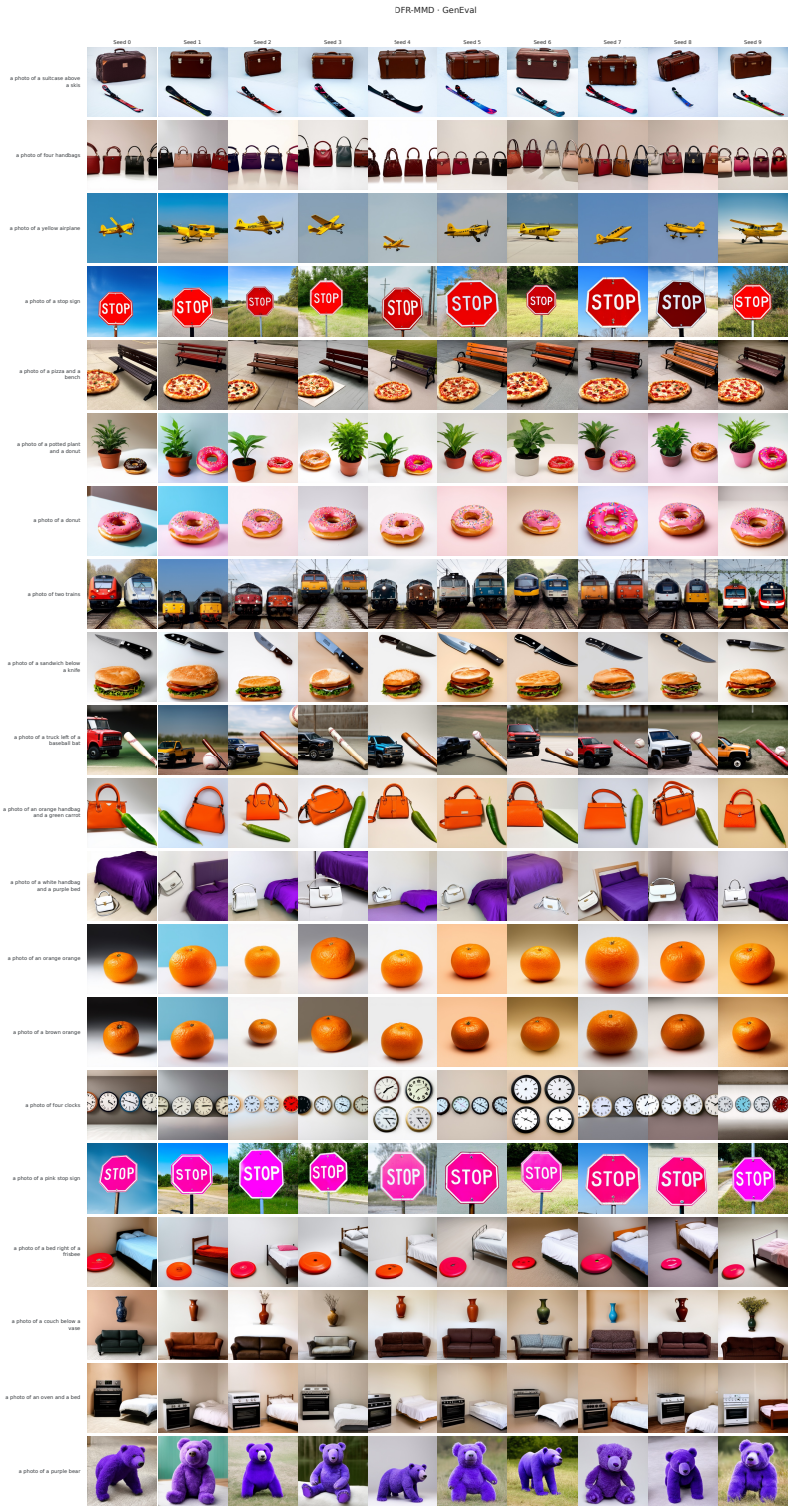


Figure 9: Sample images generated for diverse prompts across various seeds using DFR.