FLOW-MAP GRPO: REINFORCEMENT LEARNING FOR FEW-STEP FLOW-MAP GENERATORS VIA ANCHORED STOCHASTIC COMPOSITION

Anonymous authors
Paper under double-blind review



Figure 1: Few-step generation with Flow-Map GRPO. Left: images generated with Mix reward on MeanFlow and sCM. Right: Base/Ours comparisons on OCR, PickScore, and GenEval, with matched sampling steps from 2 to 8.

ABSTRACT

Few-step flow-map generators, such as consistency models and MeanFlow, accelerate sampling by learning long-range transport maps between noise and data. However, their deterministic transitions do not directly provide the stochastic trajectories and tractable likelihood ratios required by reinforcement learning (RL) post-training. Existing SDE-based stochasticization techniques target velocity-based samplers and do not directly extend to long-range flow-map transitions. We propose **Flow-Map GRPO**, an online RL post-training framework for deterministic few-step flow-map generators. Its key component, **Anchored Stochastic Flow Map Composition (ASFMC)**, combines deterministic transport with anchor-based conditional resampling. We establish the conditions under which this construction preserves the marginal probability path and develop tractable local- and endpoint-anchor policies for two-time and single-time flow maps. These policies enable a unified GRPO training procedure. Experiments on FLUX-based MeanFlow and sCM generators demonstrate substantial improvements in OCR, PickScore, and GenEval at different numbers of inference steps, including joint OCR–PickScore gains with mixed rewards. Controlled ablations show that the stochastic transition design is essential for translating training rewards into generation quality. Flow-Map GRPO enables effective RL alignment of pretrained deterministic flow-map generators while retaining their original parameterization, without retraining them as native stochastic models.

1 INTRODUCTION

Diffusion and continuous-time flow models have become a powerful framework for image generation (Ho et al., 2020; Rombach et al., 2022; Lipman et al., 2022; Liu et al., 2023). Their sampling procedures typically integrate learned dynamics over many steps. Few-step flow-map generators instead learn long-range mappings between noise and data, amortizing this integration into one or a few network evaluations. Representative methods include consistency models, shortcut models, and MeanFlow (Song et al., 2023; Lu & Song, 2025; Frans et al., 2025; Geng et al., 2025a). These

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.

models offer an attractive starting point for efficient generation, but adapting them to downstream rewards remains challenging. The difficulty is that fast deterministic sampling does not directly define a stochastic policy. Online reinforcement learning methods such as DDPO and Flow-GRPO explore alternative generations through stochastic trajectories and optimize their likelihood ratios (Black et al., 2024; Liu et al., 2026). A deterministic flow-map transition is a point mass, so it cannot supply the nondegenerate transition density needed by these methods. For velocity-based generators, an SDE reformulation provides a local Gaussian sampling rule. A long-range flow map, however, represents an entire transport interval. Adding Gaussian noise to its output does not generally reproduce the corresponding finite-time SDE transition, and resolving that SDE with many small steps would sacrifice the generator's sampling advantage.

We address this problem at the level of the flow map itself. Our key construction, **Anchored Stochastic Flow Map Composition (ASFMC)**, combines long-range deterministic transport with anchor-based conditional resampling to construct a stochastic policy for RL post-training. Given a deterministic transition, ASFMC uses an auxiliary anchor to define a resampling step with a tractable density. A local anchor leads to a Gaussian drift correction around the transported sample; an endpoint anchor uses a predicted clean sample and the model's noising schedule. ASFMC supports both MeanFlow-style two-time maps and sCM-style endpoint models without changing their network parameterization. Building on ASFMC, we introduce **Flow-Map GRPO**, which optimizes stochastic flow-map policies using group-relative rewards. The framework supports both individual-image and image-set rewards, enabling reward-based post-training while retaining the generator's original few-step sampling capability. To our knowledge, Flow-Map GRPO is the first likelihood-ratio RL method for pretrained MeanFlow generators and the first unified RL framework spanning pretrained one-time and two-time flow-map generators, enabled by stochastic policy construction directly at the flow-map level.

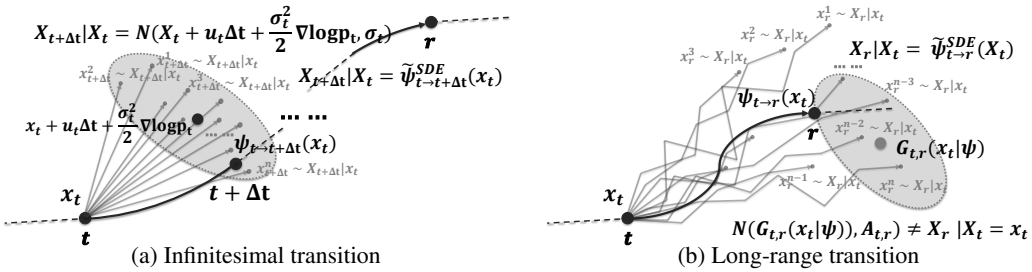
Empirically, ablations on ImageNet demonstrate the importance of the proposed stochastic policy construction for effective post-training of few-step generators, while experiments on FLUX-based MeanFlow and sCM show substantial gains under both single-reward and mixed-reward objectives.
Our contributions are as follows. (1) We identify the mismatch between deterministic long-range flow maps and likelihood-ratio post-training, and introduce ASFMC as a tractable stochasticization mechanism. (2) We develop local and endpoint policies and a unified GRPO training procedure for pretrained two-time and one-time flow-map generators. (3) We validate the framework through controlled ImageNet ablations and text-to-image experiments on FLUX-based MeanFlow and sCM under both single-reward and mixed-reward objectives.

2 BACKGROUND

Flow models and flow maps. Let $\mathcal{D} = \{x^i\}_{i=1}^n$ denote samples from an unknown data distribution $p_1 = p_{\text{data}}$ on $\mathcal{X} \subseteq \mathbb{R}^d$. A continuous-time generative model constructs a process $\{X_t\}_{t\in[0,1]}$ whose marginals evolve from a simple base distribution p_0, typically a standard Gaussian, to p_1. These marginals form a probability path $\{p_t\}_{t\in[0,1]}$. A deterministic probability-flow ODE transports samples from p_0 to p_1 along this path:

$$\frac{dx_t}{dt} = u_t(x_t), \qquad x_0 \sim p_0 = \mathcal{N}(0, I). \tag{1}$$

whose marginal densities satisfy the continuity equation $\partial_t p_t(x) + \nabla_x \cdot \big(p_t(x) u_t(x)\big) = 0$. Many flow generators use the affine path $X_t = a_t X_1 + b_t X_0$, where $X_0 \sim \mathcal{N}(0, I)$ and $X_1 \sim p_1$ are independent, with $(a_0, b_0) = (0, 1)$ and $(a_1, b_1) = (1, 0)$. The Flow Matching path is recovered by choosing $(a_t, b_t) = (t, 1-t)$, while the DDIM path is recovered by choosing $(a_t, b_t) = (\sqrt{1-(1-t)^2}, 1-t)$. The generation condition c is omitted when unambiguous. The solution of Equation (1) defines a time-dependent flow $\psi_t : \mathcal{X} \to \mathcal{X}$ with $\psi_0(x) = x$, such that $x_t = \psi_t(x_0)$, and the flow map is defined as the long-range transition between any two times t and r, $\psi_{t\to r} = \psi_r \circ \psi_t^{-1}$. Unlike multi-step methods that learn a velocity field u_t^θ and generate samples x_1 by numerically integrating Equation (1), flow-map models learn this long-range transport directly. For example, MeanFlow parameterizes the flow map through an average velocity, $\psi_{t\to r}^\theta(x_t) = x_t + (r-t) u_{t\to r}^\theta(x_t)$, and enables efficient few-step generation, such as two-step sampling $x_{0.5} = x_0 + 0.5 u_{0\to 0.5}^\theta(x_0)$ and $x_1 = x_{0.5} + 0.5 u_{0.5\to 1}^\theta(x_{0.5})$, with $x_0 \sim p_0$.



(a) Infinitesimal transition (b) Long-range transition

Figure 2: Stochasticization at different transition scales. A local SDE discretization gives a tractable Gaussian transition for an infinitesimal velocity step. Over a finite interval, accumulated nonlinear stochastic dynamics generally produce a non-Gaussian transition. This motivates constructing a stochastic policy directly around the long-range flow map.

Reward-based post-training. Recent works post-train diffusion and flow models through reinforcement learning with differentiable or non-differentiable rewards, including human preference, aesthetic quality, and downstream metrics (Liu et al., 2026; Li et al., 2025; Xue et al., 2025; Black et al., 2024). Generation is formulated as a multi-step Markov decision process (MDP). On a sampling grid $0 = t_0 < \cdots < t_N = 1$, the state is $s_i = (c, t_i, x_{t_i})$, the action is the next sample $a_i = x_{t_{i+1}}$, and the policy is $\pi_\theta(a_i \mid s_i) = p_\theta(x_{t_{i+1}} \mid x_{t_i}, c)$. The next state is (c, t_{i+1}, a_i), and a terminal reward $r(x_1, c)$ evaluates the sample. Policy-gradient methods optimize the model through its transition densities. Stochastic policies support exploration and allow likelihood ratios to be computed for policy updates and reference-policy regularization. DDPO uses stochastic diffusion denoising transitions (Black et al., 2024); Flow-GRPO introduces stochasticity into velocity-based flow sampling through a marginal-preserving SDE (Liu et al., 2026). For the ODE in Equation (1), these dynamics take the form $dX_t = [u_t(X_t) + \frac{\sigma_t^2}{2}\nabla_x \log p_t(X_t)]\, dt + \sigma_t\, dW_t$, where $q_t(x) := \nabla_x \log p_t(x)$ is the marginal score, W_t is a Wiener process, and σ_t controls the noise. The score correction preserves the marginal probability path in continuous time. This is important for introducing stochastic exploration while keeping the sampled states consistent with the pretrained model's noise levels.

3 Method

3.1 Challenge for Flow Maps

For RL post-training, as with the multi-step models in Flow-GRPO (Liu et al., 2026), flow maps need to be randomized to support stochastic exploration while preserving the pretrained marginal probability path. Specifically, if $X_t \sim p_t$, the randomized flow map $\widetilde{\psi}_{t\to r}$ should satisfy $\widetilde{\psi}_{t\to r}(X_t) \sim p_r$. The transition ablations (Section 4.1) and midpoint-anchor results (Appendix A.4) examine the practical importance of this requirement. A natural idea is to follow Flow-GRPO and inject stochasticity through an equivalent SDE whose marginal distributions match those of the deterministic ODE. For a small time step, its transition is approximated by the Gaussian update

$$X_{t+\Delta t} \approx X_t + \left[u_t(X_t) + \frac{\sigma_t^2}{2}\nabla_x \log p_t(X_t)\right]\Delta t + \sigma_t\sqrt{\Delta t}\,\epsilon_t, \qquad \epsilon_t \sim \mathcal{N}(0, I). \tag{2}$$

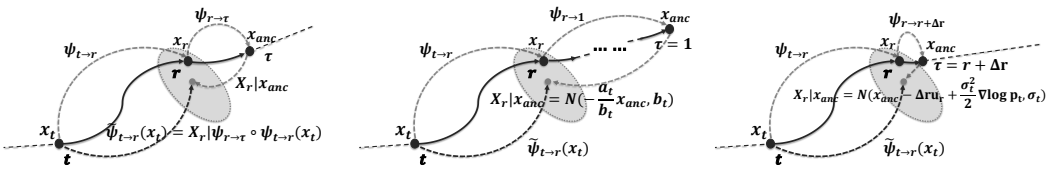
This allows stochasticity to be injected into velocity-based samplers, as in Flow-GRPO. However, flow maps are long-range transport operators between arbitrary time pairs, and their transitions are not infinitesimal. The long-range transition induced by the SDE cannot be reduced to a simple additive-noise form: a deterministic part plus an input-independent Gaussian perturbation (Fig. 2).

Theorem 3.1 (Obstruction to additive-Gaussian SDE transitions)**.** *For nonlinear drift, the exact SDE transition over $t < r$ need not admit a representation*

$$X_r = G_{t,r}(X_t; \psi, \mathrm{d}\psi, \ldots) + A_{t,r}\epsilon, \qquad \epsilon \sim \mathcal{N}(0, I), \quad \epsilon \perp X_t, \tag{3}$$

where $G_{t,r}(X_t; \psi, \mathrm{d}\psi, \ldots)$ is a deterministic function of the initial state X_t, the flow-map family ψ, and its derivatives, denoted by $\mathrm{d}\psi, \ldots$. The matrix $A_{t,r}$ is deterministic and independent of X_t. In particular, smooth additive-noise diffusions exist whose conditional transition laws are non-Gaussian, ruling out any such universal representation based on the deterministic flow map and its derivatives (see Appendix A.1 for proof).

The exact transition satisfies $X_r = X_t + \int_t^r \beta_s(X_s)\, ds + \int_t^r \sigma_s\, dW_s$, where $\beta_s = u_s + \frac{1}{2}\sigma_s^2 q_s$; its drift integral depends on the stochastic path. Resolving that trajectory with many small steps would



(a) ASFMC principle (b) Endpoint anchor (c) Local anchor

Figure 3: Anchored stochastic composition combines deterministic transport with conditional resampling. The construction transports to an anchor and samples back to the target time. A nearby anchor yields a local drift correction, while a clean endpoint prediction supplies the anchor for scheduled re-noising. Both instantiations define tractable Gaussian policies for the same GRPO update.

sacrifice the few-step advantage. As a result, the SDE-based stochasticization used in Flow-GRPO cannot be directly transferred to flow maps, which calls for a different stochasticization mechanism tailored to long-range flow-map transitions.

3.2 Anchored Stochastic Flow Map Composition

Our Anchored Stochastic Flow Map Composition (ASFMC) addresses this problem by transporting the deterministic output at time r to an anchor at time τ, then sampling back to r from the conditional distribution $p_{r|\tau}$. Specifically, it defines the randomized flow map $\widetilde{\psi}_{t\to r}(X_t) = \widetilde{X}_r$ through

$$X_r = \psi_{t\to r}(X_t), \qquad X_{\text{anc}} = X_\tau = \psi_{r\to\tau}(X_r), \qquad \widetilde{X}_r \sim p_{r|\tau}(\cdot \mid X_\tau). \tag{4}$$

The deterministic segment $r \to \tau$ produces an anchor with marginal p_τ, and the conditional transition $\tau \to r$ introduces randomness while returning to the target marginal p_r.

Theorem 3.2 (Path preservation of ASFMC)**.** *Let $X_t \sim p_t$ and define $X_r = \psi_{t\to r}(X_t)$ and $X_\tau = \psi_{r\to\tau}(X_r)$ using accurate flow maps. Let $p_{r|\tau}$ be a reverse conditional distribution whose joint law has marginals p_r and p_τ. If $\widetilde{X}_r \sim p_{r|\tau}(\cdot \mid X_\tau)$, then $\widetilde{X}_r \sim p_r$. Thus, the stochastic flow map induced by ASFMC preserves the marginal distribution of $\psi_{t\to r}$. (See Appendix A.2 for proof.)*

We describe two anchor choices below. The transition ablations (Section 4.1) and midpoint-anchor results (Appendix A.4) demonstrate the practical importance of preserving the target marginal.

Local anchor. The local anchor chooses $\tau = r + h$ for small $h > 0$, with $r + h < 1$. Over this short interval, the reverse-SDE conditional transition from the anchor X_τ to time r admits a local Gaussian approximation. To sample back from the anchor, we reverse the path-preserving SDE, obtaining the drift $u_s - \frac{1}{2}\sigma_s^2 q_s$ with s running from τ down to r (Appendix A.2). We choose $\sigma_s^2 = 2\lambda^2 b_s(\dot{a}_s b_s - a_s \dot{b}_s)/a_s$, which recovers the noise schedule of Liu et al. (2026) for the linear path, up to the noise-scale parameterization. Discretizing this reverse step and substituting the affine-path score identity gives

$$\begin{aligned}\widetilde{X}_r^\star &= X_\tau - h\left[u_\tau(X_\tau) - \tfrac{1}{2}\sigma_\tau^2 q_\tau(X_\tau)\right] + \sigma_\tau\sqrt{h}\,\xi + O_{L^2}(h^{3/2}) \\ &= \left(1 - h\lambda^2\frac{\dot{a}_\tau}{a_\tau}\right)X_\tau - h(1-\lambda^2)u_\tau(X_\tau) + \sigma_\tau\sqrt{h}\,\xi + O_{L^2}(h^{3/2}),\end{aligned} \tag{5}$$

where $\xi \sim \mathcal{N}(0, I)$. For the deterministic segment $r \to \tau$, Equation (1) gives $X_\tau = \psi_{r\to r+h}(X_r) = X_r + h u_r(X_r) + O(h^2)$, with $X_r = \psi_{t\to r}(X_t)$. Substituting this expansion and evaluating the smooth coefficients at r cancels the leading forward and reverse velocity terms. Using model predictions yields the local-anchor stochastic flow map

$$\widetilde{\psi}^{\text{loc}}_{t\to r}(X_t) = \widetilde{X}_r = X_r + h\lambda^2\left[u_r^\theta(X_r) - \frac{\dot{a}_r}{a_r}X_r\right] + \sigma_r\sqrt{h}\,\xi, \qquad \xi \sim \mathcal{N}(0, I), \tag{6}$$

where $X_r = \psi^\theta_{t\to r}(X_t)$ and u_r^θ is the instantaneous-velocity prediction. For a velocity-parameterized flow map $\psi^\theta_{t\to r}(x) = x + (r-t)u^\theta_{t\to r}(x)$, this prediction is obtained directly by equal-time evaluation, $u_r^\theta(x) = u^\theta_{r\to r}(x)$. The anchor construction thus yields a drift correction paired with Gaussian noise. The following theorem bounds the local approximation error. See Appendix A.2 for proof.



Figure 4: MeanFlow before (top) and after (bottom) joint OCR–PickScore post-training. Prompt: "A weathered treasure map laid out on an old wooden table, with 'X Marks the Spot' clearly visible in the center, surrounded by intricate illustrations of mountains, forests, and a distant coastline, all under the warm glow of a vintage lamp." Lettering, material detail, and lighting are improved

Theorem 3.3 (Error of the local-anchor approximation)**.** *Let $X_r = \psi_{t\to r}(X_t)$ and $X_{r+h} = \psi_{r\to r+h}(X_r)$ use exact transport. Let $\widetilde{X}_r^\star$ be the exact reverse-SDE sample from X_{r+h} to time r, and let $\widetilde{X}_r^{\mathrm{loc}}$ be Equation* (6) *with exact velocity. Couple both samples using the same Brownian increment, $\xi = W_h/\sqrt{h}$. Under the smoothness and moment assumptions in Appendix A.2,*

$$\|\widetilde{X}_r^\star - \widetilde{X}_r^{\mathrm{loc}}\|_{L^2} = O(h^{3/2}) \qquad \text{as } h \to 0. \tag{7}$$

Endpoint anchor. The endpoint anchor chooses $\tau = 1$. Starting from $X_r = \psi^\theta_{t\to r}(X_t)$, the anchor is obtained by transporting X_r to the data endpoint, $X_1 = \psi^\theta_{r\to 1}(X_r)$. For the affine probability path, $X_r = a_r X_1 + b_r X_0$, with $X_0 \sim \mathcal{N}(0, I)$ independent of X_1. The endpoint conditional therefore has the closed form $p_{r|1}(\cdot \mid X_1) = \mathcal{N}(a_r X_1, b_r^2 I)$. Resampling from this conditional gives

$$\widetilde{\psi}^{\mathrm{end}}_{t\to r}(X_t) = a_r \psi^\theta_{r\to 1}\big(\psi^\theta_{t\to r}(X_t)\big) + b_r \xi, \qquad \xi \sim \mathcal{N}(0, I). \tag{8}$$

Unlike the local anchor, it resamples from the endpoint conditional rather than using a short local update. Its quality consequently depends on the accuracy of the long-range endpoint transport $\psi^\theta_{r\to 1}$. Both the local- and endpoint-anchor constructions yield Gaussian policies with explicit means and variances, allowing the same likelihood-ratio training procedure.

Midpoint anchor. We use *midpoint anchor* for an anchor time $r < \tau < 1$, without requiring the arithmetic midpoint. Unlike an endpoint anchor, it retains information about both data and noise. The following proposition explains why direct Gaussian re-noising does not suffice.

Proposition 3.4 (Posterior constraint for midpoint anchors)**.** *Let $X_s = a_s X_1 + b_s X_0$, where $X_0 \sim \mathcal{N}(0, I)$ is independent of X_1. For $r < \tau < 1$ with $a_\tau \neq 0$, define $c_{r,\tau} = b_r - a_r b_\tau / a_\tau$. Then*

$$\frac{a_r}{a_\tau} x + c_{r,\tau} X_0^{\mathrm{post}} \sim p_{r|\tau}(\cdot \mid x), \qquad X_0^{\mathrm{post}} \sim p_{0|\tau}(\cdot \mid x). \tag{9}$$

If $c_{r,\tau} \neq 0$, replacing X_0^{post} with independent standard Gaussian noise is exact if and only if $p_{0|\tau}(\cdot \mid x) = p_0$. For the linear path and data with finite second moment, this substitution fails to recover the conditional kernel for every $0 \le r < \tau < 1$. (see Appendix A.4 for proof.)

A midpoint anchor thus requires posterior sampling or an approximation to it; simply reusing the endpoint noise rule omits this dependence. Our local and endpoint constructions provide tractable alternatives without introducing a separate posterior sampler. Appendix A.4 reports the corresponding midpoint-anchor experiment: OCR gains remain small while all five DrawBench metrics degrade (Tab. 5), and stochastic training rollouts exhibit visible artifacts (Fig. 6).

3.3 Flow-Map GRPO

We divide sampling into K stochastic transitions and one terminal deterministic transition, using $0 = r_0 < \cdots < r_K = r_{\mathrm{end}} < r_{K+1} = 1$. The state at step j is $s_j = (c, r_{j-1}, r_j, x_{r_{j-1}})$, the action is

x_{r_j}, and the transition policy is $\pi_\theta(x_{r_j} \mid s_j) = \mathcal{N}\big(\mu_j^\theta(s_j), \nu_j^2 I\big)$. Writing $X_{r_j} = \psi^\theta_{r_{j-1}\to r_j}(x_{r_{j-1}})$, the local anchor gives $\mu_j^\theta = X_{r_j} + h_j\lambda^2[u^\theta_{r_j}(X_{r_j}) - (\dot a_{r_j}/a_{r_j})X_{r_j}]$ and $\nu_j^2 = \sigma_{r_j}^2 h_j$, where $h_j = 0.01(r_j - r_{j-1})$ is the local anchor interval (Equation (6)). The endpoint anchor gives $\mu_j^\theta = a_{r_j}\psi^\theta_{r_j\to 1}(X_{r_j})$ and $\nu_j^2 = b_{r_j}^2$ (Equation (8)). After the stochastic steps, we apply the final endpoint map and decode the image. The reward is computed on the final decoded image, and the deterministic terminal transition is excluded from the policy-ratio terms.

For each condition c, we generate G trajectories and compute their rewards R_i. The normalized advantage is $A_i = (R_i - \mu_c)/(\sigma_{\text{global}} + \epsilon_{\text{std}})$, and the clipped advantage is $\widehat{A}_i = \text{clip}(A_i, -A_{\max}, A_{\max})$. Here $\mu_c = \frac{1}{G}\sum_{k=1}^G R_k$. The quantity σ_{global} is the reward standard deviation over the rollout batch, ϵ_{std} stabilizes the denominator, and $A_{\max}$ is the advantage clipping threshold; Appendix C.2 specifies the statistics and settings. For trajectory i at step j, the likelihood ratio is $\rho_{ij} = \pi_\theta(x_{r_j}^{(i)} \mid s_j^{(i)})/\pi_{\theta_{\text{old}}}(x_{r_j}^{(i)} \mid s_j^{(i)})$, where θ_{old} denotes the policy parameters used to collect the rollout. The clipped GRPO objective is

$$J(\theta) = \mathbb{E}\left[\frac{1}{GK}\sum_{i=1}^{G}\sum_{j=1}^{K}\min\Big(\rho_{ij}\widehat{A}_i, \text{clip}(\rho_{ij}, 1-\epsilon, 1+\epsilon)\widehat{A}_i\Big)\right]. \tag{10}$$

We update LoRA adapters while keeping the pretrained weights frozen. Actions and rewards are fixed during each update. Appendix A.3 specifies the likelihood normalization and gradient conventions in optimization.

Algorithm 1 summarizes the procedure. The endpoint anchor also applies to one-time models, which fix the target time to the data endpoint 1 and learn $f_t^\theta(x_t) = \psi^\theta_{t\to 1}(x_t)$. For the endpoint anchor, accurate flow maps satisfy $\psi_{r\to 1} \circ \psi_{t\to r} = \psi_{t\to 1}$. The one-time model therefore predicts the result of these two transport segments directly, supplying $X_1 = f^\theta_{r_{j-1}}(x_{r_{j-1}})$ in Equation (8). Its Gaussian policy therefore uses $\mu_j^\theta = a_{r_j} f^\theta_{r_{j-1}}(x_{r_{j-1}})$ and $\nu_j^2 = b_{r_j}^2$, and is optimized with the same GRPO objective. The parameter settings for MeanFlow and sCM are provided in Appendix C.2. Evaluation can use 3 to 16 sampling steps, independently of the $K+1$ steps during training (Section 4).

Algorithm 1 Flow-Map GRPO with ASFMC

Require: Generator θ, grid $\{r_j\}$, group size G, reward R, learning rate η_{opt}

while training steps remain **do**
  $\theta_{\text{old}} \leftarrow \theta$; sample conditions c
  for each c and $i = 1, \dots, G$ **do**
    Sample $x_{r_0}^{(i)} \sim \mathcal{N}(0, I)$
    for $j = 1, \dots, K$ **do**
      Form state $s_j^{(i)}$
      if local policy **then**
        Predict transport and local velocity
        Obtain (μ_j, ν_j) from (6)
      else
        Predict endpoint X_1
        $\mu_j = a_{r_j} X_1$, $\nu_j = b_{r_j}$
      end if
      Sample $x_{r_j}^{(i)} \sim \mathcal{N}(\mu_j, \nu_j^2 I)$
      Store $s_j^{(i)}$, $x_{r_j}^{(i)}$, $\log \pi_{\theta_{\text{old}}}(x_{r_j}^{(i)} \mid s_j^{(i)})$
    end for
    Predict final endpoint; decode I_i
  end for
  Compute rewards R_i and advantages $\widehat{A}_i$
  Recompute $\log \pi_\theta(x_{r_j}^{(i)} \mid s_j^{(i)})$
  $\theta \leftarrow \theta + \eta_{\text{opt}} \nabla_\theta J(\theta)$ using (10)
end while

Flow-Map GRPO supports both image-level and image-set rewards, whether differentiable or non-differentiable, because policy-gradient optimization does not require gradients through the reward. For image-level rewards such as OCR, PickScore, and GenEval, each trajectory receives the score of its generated image, $R_i = r(I_i \mid c)$. For an image-set reward $\mathcal{R}(\mathbb{X}_g; \mathbb{X}_r)$, where $\mathbb{X}_g = \{I_1, \dots, I_G\}$ is the generated group and $\mathbb{X}_r$ is a reference set for condition c, we follow Miao et al. (2024) and assign each trajectory its marginal contribution, $R_i = \mathcal{R}(\mathbb{X}_g; \mathbb{X}_r) - \mathcal{R}(\mathbb{X}_g \setminus \{I_i\}; \mathbb{X}_r)$. Both reward types then use the same advantage computation and GRPO objective. Our ImageNet experiments use the negative MMD-squared estimate as an image-set reward.

4 Experiments

We conduct ablation studies on ImageNet and evaluate Flow-Map GRPO on FLUX-based MeanFlow and sCM text-to-image generators with single and mixed rewards. We use K for the number of transitions before the final endpoint step, so sampling takes $K+1$ steps. Training uses $K = 4$, corresponding to five sampling steps. In the appendix, we also report results at other inference step counts, demonstrating flexibility to use different numbers of steps during training and inference.

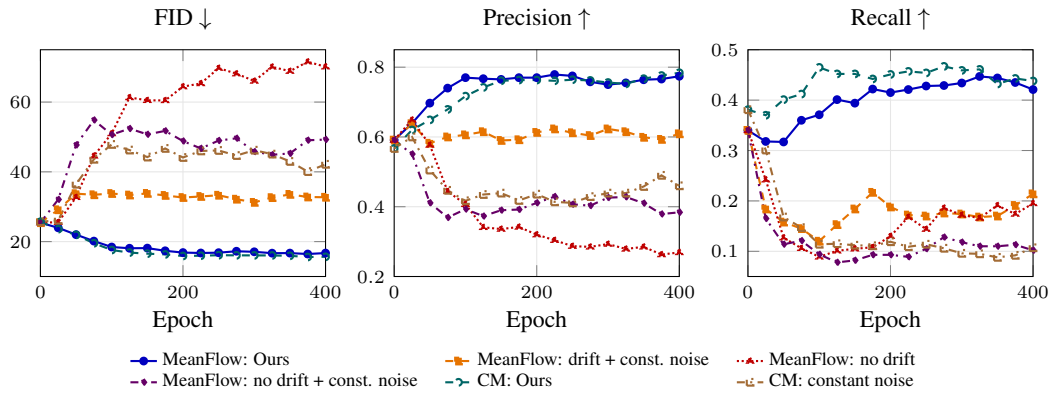


Figure 5: Learning dynamics of Ours and ablation variants, measured by FID, Precision, and Recall. MeanFlow compares Ours with No drift, Drift + const. noise, and No drift + const. noise; CM compares Ours with constant noise.

4.1 Ablation Study

We post-train MeanFlow and CM on ImageNet using LoRA adaptation and the maximum mean discrepancy (MMD) image-set reward of Miao et al. (2024). All variants use the same training and evaluation settings. Appendix C.1 gives the configurations. We evaluate generation quality using Fréchet Inception Distance (FID), Precision, and Recall. Lower FID indicates closer agreement between generated and real-image distributions, while higher Precision and Recall indicate better sample fidelity and broader coverage of the real-image distribution, respectively.

Local-anchor ablation study. We ablate the drift correction $h\lambda^2[u_r^\theta(X_r) - (\dot{a}_r/a_r)X_r]$ and noise scale $\sigma_r\sqrt{h}$ in Equation (6), keeping the transport and training settings fixed. *Ours* retains both terms as derived from the local anchor. *Drift + const. noise* retains the drift correction but replaces the scheduled noise scale with a constant matched to the total injected variance. *No drift* removes the entire drift correction while retaining the scheduled noise. *No drift + const. noise* removes the drift correction and uses the same variance-matched constant noise scale. The two no-drift variants thus directly add Gaussian noise to the deterministic MeanFlow output, using scheduled or constant noise, respectively. Only the correct form derived from the local anchor improves FID, Precision, and Recall together (Tab. 1), supporting the necessity of our method. Simply adding noise to the deterministic flow-map output is insufficient. See Fig. 5 for learning dynamics.

Table 1: Five-step ablations at epoch 400. MeanFlow uses deterministic sampling; CM uses native stochastic sampling. Each uses its own Base.

Policy	FID ↓	Prec. ↑	Rec. ↑
MeanFlow			
Base	25.43	0.591	0.340
Ours	**16.72**	**0.774**	**0.421**
Drift + const. noise	32.71	0.609	0.214
No drift	70.13	0.269	0.195
No drift + const. noise	49.25	0.384	0.102
CM			
Base	25.69	0.567	0.381
Ours	**15.65**	**0.784**	**0.438**
Constant noise	42.16	0.460	0.107

Endpoint-anchor ablation study. We replace the noise coefficient b_r of our endpoint-anchor policy in Equation (8) with a variance-matched constant, keeping the mean $a_r X_1$ unchanged. Our endpoint-anchor policy (*Ours*) improves fidelity and coverage under the native CM sampler, whereas constant noise reverses these gains, supporting the necessity of the correct endpoint-anchor formulation. The learning dynamics are shown in Fig. 5.

4.2 Single-Score Optimization

Rewards. We post-train FLUX.1-lite MeanFlow and sCM (Pu et al., 2025) separately with OCR and PickScore rewards. OCR measures text-rendering accuracy, and PickScore measures human image preference. Training uses five sampling steps; Appendix C.2 gives the settings. Appendix C.3 motivates the local-anchor choice for MeanFlow in this T2I setting.

Table 2: OCR-only and PickScore-only post-training at $K = 4$. Each reward block compares Base with a separately trained Flow-Map GRPO adapter. Task Metrics measure RL improvements on the target tasks using their respective test sets. DrawBench Metrics are auxiliary general-quality metrics used to monitor degradation outside the optimized reward. Evaluation details are in Appendix C.2.

Model	Reward	Task Metrics		DrawBench Metrics				
		OCR Acc.↑	PickScore Task↑	DB-PickScore↑	Aesthetic↑	DeQA↑	ImgRwd↑	UniRwd↑
MeanFlow Base	OCR	0.3491	–	21.6298	**5.5368**	**4.1712**	0.3918	2.7187
MeanFlow + Flow-Map GRPO LoRA	OCR	**0.8504**	–	**21.9869**	5.5111	4.1396	**0.5386**	**2.8049**
MeanFlow Base	PickScore	–	20.9758	21.6298	5.5368	**4.1712**	0.3918	2.7187
MeanFlow + Flow-Map GRPO LoRA	PickScore	–	**23.0772**	**23.3711**	**6.1510**	4.0829	**1.1402**	**3.2005**
sCM Base	OCR	0.4903	–	21.4993	**5.4747**	3.7469	**0.5750**	2.6725
sCM + Flow-Map GRPO LoRA	OCR	**0.8863**	–	**21.5647**	5.4049	**3.7800**	0.3475	**2.7137**
sCM Base	PickScore	–	20.3517	21.4993	5.4747	3.7469	0.5750	2.6725
sCM + Flow-Map GRPO LoRA	PickScore	–	**23.4820**	**23.8084**	**6.0360**	**4.2343**	**1.2600**	**3.3361**

Results. Both models improve substantially on both target rewards (Tab. 2). At five steps, OCR accuracy exceeds 0.85 for both models, and PickScore increases by more than two points. PickScore optimization also improves all five DrawBench metrics for sCM and four for MeanFlow; OCR optimization shows mixed effects on these auxiliary metrics. Appendices D.1 and D.2 show that the target-reward gains persist at every tested inference step count, demonstrating transfer from five-step training to other inference step counts without retraining.

Comparative experiments. In the absence of an established likelihood-ratio RL baseline for pretrained MeanFlow generators, Tab. 3 compares SFT and DiffusionNFT as representative existing post-training approaches. We use the same pretrained models, prompt splits, reward models, sampling settings, and number of optimizer steps; Appendix C.4 details the baseline objectives and matched comparison settings. Flow-Map GRPO achieves the highest reward in all four comparisons. SFT and DiffusionNFT yield limited gains or reduce the target reward. Flow-map structure creates two challenges for these objectives: native supervision couples long-range predictions to an unadapted instantaneous field, while forward-noising regression requires local conditional predictions that flow-map outputs do not generally represent. Appendix C.4 explains these objective mismatches; Appendix C.5 shows the training curves.

Table 3: Single-reward post-training at $K = 4$.

Method	sCM		MeanFlow	
	OCR↑	PickScore↑	OCR↑	PickScore↑
Base	0.4903	20.3517	0.3491	20.9758
SFT	0.4588	20.7893	0.3360	20.9763
NFT	0.5363	20.2200	0.3576	20.5418
Ours	**0.8863**	**23.4820**	**0.8503**	**23.0772**

4.3 Mixed-Score Optimization

We evaluate OCR–PickScore mixed-reward optimization and GenEval compositional-reward optimization, using a separate adapter for each objective on each backbone. For OCR–PickScore, we use a mixture of OCR and PickScore prompts. For prompts with an OCR target, we use $R_{\text{mix}} = 0.7\widetilde{R}_{\text{OCR}} + 0.3\widetilde{R}_{\text{Pick}}$; otherwise, we use $\widetilde{R}_{\text{Pick}}$ alone. The $\tilde{}$ denotes reward normalization. GenEval (Ghosh et al., 2023) rewards satisfaction of the objects, counts, colors, and spatial relations specified by the prompt. Appendix C.2 gives the reward definitions and training configurations.

Results. Tab. 4 shows that OCR–PickScore mixed-reward optimization improves both target metrics on both backbones, while GenEval optimization substantially improves compositional accuracy. Fig. 4 illustrates the mixed-reward gains in text rendering and visual detail. Mixed-reward optimization also yields broad gains or comparable performance across the DrawBench metrics and inference step counts. GenEval optimization shows mixed effects on the auxiliary quality metrics. The mixed-reward adapters improve both tasks jointly, although their individual task scores remain below those of the corresponding single-reward adapters. Appendices D.3 and D.4 show that target-reward improvements transfer to other inference step counts.

5 Related Work

Few-step generative models. Consistency models and sCM learn endpoint predictors for efficient generation (Song et al., 2023; Song & Dhariwal, 2024; Lu & Song, 2025). Shortcut models, Flow

Table 4: GenEval and joint OCR–PickScore post-training at $K = 4$. The objectives use separate adapters. Joint training combines OCR and PickScore prompts with the reward mixture described in Section 4.3. Task Metrics measure RL improvements on the target tasks using their test sets. DrawBench Metrics are general-quality metrics used to monitor degradation outside the optimized reward.

Model	Reward	Task Metrics			DrawBench Metrics				
		OCR Acc.↑	PickScore Task↑	GenEval↑	DB-PickScore↑	Aesthetic↑	DeQA↑	ImgRwd↑	UniRwd↑
MeanFlow Base	GenEval	–	–	0.5041	21.6298	**5.5368**	**4.1712**	0.3918	2.7187
MeanFlow + Flow-Map GRPO LoRA	GenEval	–	–	**0.8454**	**21.8184**	5.3905	3.2086	**0.6426**	**2.8719**
sCM Base	GenEval	–	–	0.4513	21.4993	**5.4747**	3.7469	0.5750	2.6725
sCM + Flow-Map GRPO LoRA	GenEval	–	–	**0.8049**	**22.1046**	5.2401	**3.8938**	**0.6514**	**3.0835**
MeanFlow Base	OCR+PickScore	0.3491	20.9758	–	21.6298	5.5368	4.1712	0.3918	2.7187
MeanFlow + Flow-Map GRPO LoRA	OCR+PickScore	**0.7719**	**22.4964**	–	**23.0506**	**5.9299**	**4.1892**	**1.0439**	**3.1876**
sCM Base	OCR+PickScore	0.4903	20.3517	–	21.4993	5.4747	3.7469	0.5750	2.6725
sCM + Flow-Map GRPO LoRA	OCR+PickScore	**0.8568**	**22.5153**	–	**23.0403**	**5.9494**	**4.3311**	**0.9741**	**3.1004**

Map Matching, and MeanFlow learn long-range transitions or average velocities (Frans et al., 2025; Boffi et al., 2025; Geng et al., 2025a;b). These methods reduce the sampling cost of diffusion and flow models through distillation or derivative-based training. Flow-Map GRPO operates after this stage: it adapts pretrained generators to downstream rewards while retaining their prediction interface.

Reinforcement learning for generation. DDPO optimizes diffusion models using stochastic denoising trajectories (Black et al., 2024). Flow-GRPO extends online RL to velocity-based flow matching through stochastic sampling, while DanceGRPO and MixGRPO further develop reward-based visual generation and efficient rollout strategies (Liu et al., 2026; Xue et al., 2025; Li et al., 2025). RLCM applies policy-gradient RL to Gaussian consistency-sampler transitions (Oertell et al., 2024), while ROCM uses direct reward gradients with distributional regularization (Shekhar & Zhang, 2025). Our contribution is the stochastic policy construction for deterministic long-range maps, where a velocity-level SDE discretization does not directly provide the desired few-step transition.

Stochastic and reward-aware flow maps. Meta Flow Maps, Diamond Maps, and Strong Stochastic Flow Maps learn stochastic mapping mechanisms for posterior sampling or reward alignment (Potaptchik et al., 2026; Holderrieth et al., 2026a; McCallum et al., 2026). Related approaches study stochastic few-step sampling, inference-time alignment, and reward guidance (Passaro et al., 2026; Holderrieth et al., 2026b; Huang et al., 2026a). Flow-Map GRPO instead constructs a post-training policy around a pretrained deterministic generator. It introduces exploration through anchored resampling, without requiring the base model to be retrained as a native stochastic flow map.

Concurrent work. Concurrent to our work, MeanFlowNFT (Huang et al., 2026b) adapts DiffusionNFT's forward-process RL to MeanFlow, using the MeanFlow identity to reconcile instantaneous-velocity supervision with average-velocity sampling. RAVEN (Lu et al., 2026) introduces CM-GRPO for video post-training through Gaussian consistency-sampler transitions. In contrast, Flow-Map GRPO constructs stochastic policies around pretrained flow-map generators and provides a likelihood-ratio RL formulation for MeanFlow-style two-time maps and consistency-style one-time predictors. See Appendix B for extended related work.

6 Conclusion

We introduced Flow-Map GRPO, a framework for online reward-based post-training of deterministic few-step flow-map generators. ASFMC supplies tractable local and endpoint policies that support GRPO updates while retaining the pretrained model parameterization. Controlled ablations show that the stochastic transition is crucial for transferring training improvements to generation, and text-to-image experiments demonstrate substantial gains in PickScore, OCR, and GenEval on MeanFlow and sCM. Joint OCR–PickScore optimization improves both target tasks with one adapter per backbone, while the auxiliary metrics reveal quality trade-offs that depend on the backbone and number of sampling steps. These results support anchored stochastic composition as an effective route to reward-based adaptation of pretrained few-step generators. Regarding limitations and future work, our empirical study focuses on image generation with MeanFlow and sCM; the text-to-image results use one training seed and a fixed reward mixture. Extending the evaluation across seeds, modalities, and reward combinations will further characterize the framework's scope. Adaptive anchor selection and resampling schedules offer a natural direction for improving the quality–cost trade-off.

AI USE STATEMENT

We used generative AI tools primarily to collect, organize, and transcribe experimental information for the appendix, and to polish the writing of the main text. We also used these tools for literature retrieval and discovery to identify relevant related work. For mathematical work, AI assisted with developing the counterexample construction for Theorem 3.1 and writing the proof of Theorem 3.3. The authors take responsibility for the final content of this paper, including the AI-assisted text, mathematical claims, and proofs.

REFERENCES

Michael Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025.

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

Nicholas Matthew Boffi, Michael Samuel Albergo, and Eric Vanden-Eijnden. Flow map matching with stochastic interpolants: A mathematical framework for consistency models. *Transactions on Machine Learning Research (TMLR)*, 2025.

Yuning Du, Chenxia Li, Ruoyu Guo, Xiaoting Yin, Weiwei Liu, Jun Zhou, Yifan Bai, Zilin Yu, Yehua Yang, Qingqing Dang, and Haoshuang Wang. PP-OCR: A practical ultra lightweight OCR system. *arXiv preprint arXiv:2009.09941*, 2020.

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *International Conference on Learning Representations (ICLR)*, 2025.

Zhengyang Geng, Ashwini Pokle, William Luo, Justin Lin, and J Zico Kolter. Consistency models made easy. *arXiv preprint arXiv:2406.14548*, 2024.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *arXiv preprint arXiv:2505.13447*, 2025a.

Zhengyang Geng, Yiyang Lu, Zongze Wu, Eli Shechtman, J Zico Kolter, and Kaiming He. Improved mean flows: On the challenges of fastforward generative models. *arXiv preprint arXiv:2512.02012*, 2025b.

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

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

Peter Holderrieth, Douglas Chen, Luca Eyring, Ishin Shah, Giri Anantharaman, Yutong He, Zeynep Akata, Tommi Jaakkola, Nicholas Matthew Boffi, and Max Simchowitz. Diamond maps: Efficient reward alignment via stochastic flow maps. *arXiv preprint arXiv:2602.05993*, 2026a.

Peter Holderrieth, Uriel Singer, Tommi Jaakkola, Ricky TQ Chen, Yaron Lipman, and Brian Karrer. Glass flows: Efficient inference for reward alignment of flow and diffusion models. In *The Fourteenth International Conference on Learning Representations*, 2026b.

Jerry Y Huang, Justin Lin, Sheel Shah, Kartik Nair, and Nicholas M Boffi. How to guide your flow: Few-step alignment via flow map reward guidance. *arXiv preprint arXiv:2604.27147*, 2026a.

Yushi Huang, Xiangxin Zhou, Jun Zhang, Liefeng Bo, and Tianyu Pang. MeanFlowNFT: Bringing forward-process RL to average-velocity generators. *arXiv preprint arXiv:2607.15273*, 2026b.

Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2305.01569.

Kimin Lee, Hao Liu, Moonkyung Ryu, Olivia Watkins, Yuqing Du, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, and Shixiang Shane Gu. Aligning text-to-image models using human feedback. *arXiv preprint arXiv:2302.12192*, 2023.

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

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

Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky TQ Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code. *arXiv preprint arXiv:2412.06264*, 2024.

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

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*, 2023.

Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models. In *International Conference on Learning Representations (ICLR)*, 2025.

Yanzuo Lu, Ronglai Zuo, and Jiankang Deng. RAVEN: Real-time autoregressive video extrapolation with consistency-model GRPO. *arXiv preprint arXiv:2605.15190*, 2026.

Sam McCallum, Zander W Blasingame, Timothy Herschell, Niklas Rindtorff, Alexander Tong, and James Foster. Strong stochastic flow maps. *arXiv preprint arXiv:2606.01086*, 2026.

Zichen Miao, Jiang Wang, Ze Wang, Zhengyuan Yang, Lijuan Wang, Qiang Qiu, and Zicheng Liu. Training diffusion models towards diverse image generation with reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10844–10853, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Miao_Training_Diffusion_Models_Towards_Diverse_Image_Generation_with_Reinforcement_Learning_CVPR_2024_paper.html.

Owen Oertell, Jonathan D. Chang, Yiyi Zhang, Kiante Brantley, and Wen Sun. Rl for consistency models: Faster reward guided text-to-image generation. *arXiv preprint arXiv:2404.03673*, 2024.

Romeo Passaro, Zander W Blasingame, Michael M Bronstein, and Alexander Tong. Stochastic few-step models. In *2nd DeLTa Workshop at ICLR*, 2026. URL https://openreview.net/pdf/28374f68a7ac1a65769d7cc4b5aee5bc1ac66351.pdf.

Peter Potaptchik, Adhi Saravanan, Abbas Mammadov, Alvaro Prat, Michael S Albergo, and Yee Whye Teh. Meta flow maps enable scalable reward alignment. *arXiv preprint arXiv:2601.14430*, 2026.

Yifan Pu, Yizeng Han, Zhiwei Tang, Jiasheng Tang, Fan Wang, Bohan Zhuang, and Gao Huang. Few-step distillation for text-to-image generation: A practical guide. *arXiv preprint arXiv:2512.13006*, 2025.

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.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2205.11487.

Christoph Schuhmann. Improved aesthetic predictor, 2022. URL https://github.com/christophschuhmann/improved-aesthetic-predictor.

Shivanshu Shekhar and Tong Zhang. ROCM: RLHF on consistency models. *arXiv preprint arXiv:2503.06171*, 2025.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020a.

Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. In *International Conference on Learning Representations (ICLR)*, 2024.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020b.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning (ICML)*, 2023.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the Inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2016. doi: 10.1109/CVPR.2016.308. URL https://openaccess.thecvf.com/content_cvpr_2016/html/Szegedy_Rethinking_the_Inception_CVPR_2016_paper.html.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. *arXiv preprint arXiv:2311.12908*, 2023.

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

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. ImageReward: Learning and evaluating human preferences for text-to-image generation. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2304.05977.

Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, et al. Dancegrpo: Unleashing grpo on visual generation. *arXiv preprint arXiv:2505.07818*, 2025.

Zhiyuan You, Xin Cai, Jinjin Gu, Tianfan Xue, and Chao Dong. Teaching large language models to regress accurate image quality scores using score distribution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. URL https://arxiv.org/abs/2501.11561.

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

APPENDIX CONTENTS

A Theoretical Details and Proofs . . . 13
A.1 Proof of the long-range SDE obstruction . . . 13
A.2 ASFMC path preservation and anchor constructions . . . 14
A.3 Gaussian likelihoods and optimization . . . 16
A.4 Midpoint anchors and the posterior constraint . . . 17
A.5 Unified Interpretation of Anchor Choices . . . 19
A.6 Evaluation Sampling for One-Time and Two-Time Models . . . 19
B Extended Related Work . . . 19
C Experiment Details . . . 21
C.1 MeanFlow and CM Ablation Details . . . 21
C.2 Single-Score and Mixed-Score Post-Training Details . . . 23
C.3 Anchor Choice for MeanFlow Text-to-Image Post-Training . . . 25
C.4 Baseline Setting and Challenges Posed by Flow Maps . . . 26
C.5 OCR and PickScore Training Curves . . . 28
D Complete Text-to-Image Results . . . 29
D.1 OCR-Only Evaluations . . . 29
D.2 PickScore-Only Evaluations . . . 30
D.3 GenEval Evaluations . . . 31
D.4 Joint OCR–PickScore Evaluations . . . 31
E Additional Qualitative Results . . . 32
E.1 MeanFlow: OCR . . . 33
E.2 sCM: OCR . . . 34
E.3 MeanFlow: PickScore . . . 35
E.4 sCM: PickScore . . . 36
E.5 MeanFlow: GenEval . . . 37
E.6 sCM: GenEval . . . 38
E.7 MeanFlow: OCR + PickScore . . . 39
E.8 sCM: OCR + PickScore . . . 40

A THEORETICAL DETAILS AND PROOFS

A.1 PROOF OF THE LONG-RANGE SDE OBSTRUCTION

Theorem 3.1 (Obstruction to additive-Gaussian SDE transitions). For nonlinear drift, the exact SDE transition over $t < r$ need not admit a representation

$$X_r = G_{t,r}(X_t; \psi, \mathrm{d}\psi, \ldots) + A_{t,r}\epsilon, \qquad \epsilon \sim \mathcal{N}(0, I), \quad \epsilon \perp X_t,$$

where $G_{t,r}(X_t; \psi, \mathrm{d}\psi, \ldots)$ is a deterministic function of the initial state X_t, the flow-map family ψ, and its derivatives, denoted by $\mathrm{d}\psi, \ldots$. The matrix $A_{t,r}$ is deterministic and independent of X_t. In particular, smooth additive-noise diffusions exist whose conditional transition laws are non-Gaussian, ruling out any such universal representation based on the deterministic flow map and its derivatives.

Proof. We exhibit a smooth nonlinear diffusion with a non-Gaussian conditional transition. Fix $X_0 = x$ and consider $dX_s = \beta(X_s)\, ds + \sigma\, dW_s$, where $\sigma > 0$ is constant and β is smooth and bounded with bounded derivatives. The choice $\beta(x) = \sin x$ satisfies these assumptions. The generator is $L = \beta\partial_x + \frac{\sigma^2}{2}\partial_{xx}$. For the polynomial test functions used below, finite moments and

the generator expansion give

$$\mathbb{E}[f(X_h) \mid X_0 = x] = f(x) + hLf(x) + \frac{h^2}{2}L^2 f(x) + \frac{h^3}{6}L^3 f(x) + o(h^3). \tag{11}$$

Write $M_k = \mathbb{E}[(X_h - x)^k \mid X_0 = x]$ and evaluate β, β', β'' at x. Applying Equation (11) to $(z-x)^k$ yields the terms needed for the third centered moment:

$$\begin{aligned} M_1 &= \beta h + \left(\tfrac{1}{2}\beta\beta' + \tfrac{1}{4}\sigma^2\beta''\right)h^2 + O(h^3), \\ M_2 &= \sigma^2 h + (\beta^2 + \sigma^2\beta')h^2 + O(h^3), \\ M_3 &= 3\sigma^2\beta h^2 + \left(\beta^3 + \tfrac{9}{2}\sigma^2\beta\beta' + \tfrac{7}{4}\sigma^4\beta''\right)h^3 + o(h^3). \end{aligned} \tag{12}$$

Consequently, with $m_h = \mathbb{E}[X_h \mid X_0 = x]$, the terms of order h^2, $\beta^3 h^3$, and $\sigma^2\beta\beta' h^3$ cancel:

$$\mathbb{E}[(X_h - m_h)^3 \mid X_0 = x] = M_3 - 3M_1M_2 + 2M_1^3 = \sigma^4\beta''(x)h^3 + o(h^3). \tag{13}$$

For $\beta(x) = \sin x$ and $x = \pi/2$, this quantity is nonzero for all sufficiently small $h > 0$. In contrast, any representation $G_{0,h}(x; \psi, \mathrm{d}\psi, \ldots) + A_{0,h}\epsilon$ with independent Gaussian ϵ has a Gaussian conditional law and zero third centered moment. No such representation can therefore hold for general long-range SDE transitions. A scalar example suffices to disprove the general claim; it can also be embedded in a higher-dimensional diffusion using independent additional coordinates.

This example is compatible with the probability-flow setting. Start the diffusion from a smooth positive initial density and let p_s be its marginal density. Its Fokker–Planck equation is $\partial_s p_s = -\partial_x(\beta p_s) + \frac{\sigma^2}{2}\partial_{xx}p_s$. Defining $u_s = \beta - \frac{\sigma^2}{2}\partial_x \log p_s$ gives $\partial_s p_s = -\partial_x(u_s p_s)$ and $\beta = u_s + \frac{\sigma^2}{2}\partial_x \log p_s$. Thus the same diffusion is a path-preserving stochasticization of its probability-flow ODE wherever the associated smooth flow is defined. The obstruction concerns a general additive-Gaussian representation; it does not assert that every nonlinear diffusion has non-Gaussian transitions at every time pair.

□

A.2 ASFMC PATH PRESERVATION AND ANCHOR CONSTRUCTIONS

Theorem 3.2 (Path preservation of ASFMC). Let $X_t \sim p_t$ and define $X_r = \psi_{t\to r}(X_t)$ and $X_\tau = \psi_{r\to\tau}(X_r)$ using accurate flow maps. Let $p_{r|\tau}$ be a reverse conditional distribution whose joint law has marginals p_r and p_τ. If $\widetilde{X}_r \sim p_{r|\tau}(\cdot \mid X_\tau)$, then $\widetilde{X}_r \sim p_r$. Thus, the stochastic flow map induced by ASFMC preserves the marginal distribution of $\psi_{t\to r}$.

Proof. Exact transport first gives $X_r \sim p_r$ and then $X_\tau \sim p_\tau$. Let $P_{r,\tau}$ be the joint law defining the selected conditional, so $P_{r,\tau}(dx_r, dx_\tau) = p_{r|\tau}(dx_r \mid x_\tau)p_\tau(dx_\tau)$ has first marginal p_r. For every measurable set A,

$$\Pr(\widetilde{X}_r \in A) = \int p_{r|\tau}(A \mid x_\tau)\, p_\tau(dx_\tau) = p_r(A). \tag{14}$$

This establishes preservation at one transition. Applying the same argument successively on a sampling grid preserves every grid-time marginal, provided each transport and conditional is exact. The joint law used for resampling may be the forward diffusion for a local anchor or the affine interpolant for an endpoint anchor; it need not be the deterministic flow's trajectory coupling. □

Affine-path velocity and score. Let $X_s = a_s X_1 + b_s X_0$, with $X_0 \sim \mathcal{N}(0, I)$ independent of X_1, as in Section 2. Work on an interior interval where $a_s, b_s > 0$ and $D_s = \dot{a}_s b_s - a_s \dot{b}_s > 0$. Assume that the path coefficients are smooth and the conditional moments below exist. Differentiating the Gaussian conditional density and averaging over X_1 gives

$$q_s(x) = \nabla_x \log p_s(x) = -\frac{x - a_s\mathbb{E}[X_1 \mid X_s = x]}{b_s^2} = -\frac{1}{b_s}\mathbb{E}[X_0 \mid X_s = x]. \tag{15}$$

The marginal velocity of the affine interpolant is $u_s(x) = \mathbb{E}[\dot{a}_s X_1 + \dot{b}_s X_0 \mid X_s = x]$. Together with $x = a_s\mathbb{E}[X_1 \mid X_s = x] + b_s\mathbb{E}[X_0 \mid X_s = x]$, this implies (Lipman et al., 2024)

$$q_s(x) = \frac{a_s u_s(x) - \dot{a}_s x}{b_s D_s}. \tag{16}$$

For Flow Matching, $(a_s, b_s) = (s, 1-s)$ gives $D_s = 1$; for the DDIM path, $(a_s, b_s) = (\sqrt{1-(1-s)^2}, 1-s)$ gives $D_s = 1/a_s$. Both are positive for $0 < s < 1$. Boundary transitions are handled by the endpoint maps rather than by substituting a zero denominator into Equation (16).

Forward and reverse stochastic dynamics. For a spatially constant diffusion coefficient σ_s, the forward drift $u_s + \frac{1}{2}\sigma_s^2 q_s$ preserves the marginal path because

$$-\nabla\cdot\left[\left(u_s + \tfrac{1}{2}\sigma_s^2 q_s\right) p_s\right] + \tfrac{1}{2}\sigma_s^2 \Delta p_s = -\nabla\cdot(u_s p_s). \tag{17}$$

Reversing this diffusion yields the drift $B_s = u_s - \frac{1}{2}\sigma_s^2 q_s$ when s runs backward. More explicitly, starting at $\widehat{X}_0 = X_\tau$ and using elapsed reverse time $v \in [0, h]$, the conditional sampler is $d\widehat{X}_v = -B_{\tau-v}(\widehat{X}_v)\, dv + \sigma_{\tau-v}\, dW_v$. Choosing $\sigma_s^2 = 2\lambda^2 b_s D_s / a_s$ yields

$$B_s(x) = (1-\lambda^2)u_s(x) + \lambda^2 \frac{\dot{a}_s}{a_s} x, \qquad \tfrac{1}{2}\sigma_s^2 q_s(x) = \lambda^2 \left[u_s(x) - \frac{\dot{a}_s}{a_s} x\right]. \tag{18}$$

Local-anchor expansion. Set $X_r = \psi_{t\to r}(X_t)$, $\tau = r + h < 1$, and $X_\tau = \psi_{r\to\tau}(X_r)$ using exact transport. Under the assumptions below, the forward segment and the reverse stochastic step satisfy

$$\begin{aligned} X_\tau &= X_r + h u_r(X_r) + O_{L^2}(h^2), \\ \widetilde{X}_r^\star &= X_\tau - h B_\tau(X_\tau) + \sigma_\tau \sqrt{h}\, \xi + O_{L^2}(h^{3/2}), \qquad \xi \sim \mathcal{N}(0, I). \end{aligned} \tag{19}$$

Since $B_\tau(X_\tau) = B_r(X_r) + O_{L^2}(h)$ and $\sigma_\tau = \sigma_r + O(h)$, substitution gives

$$\begin{aligned} \widetilde{X}_r^\star &= X_r + h[u_r(X_r) - B_r(X_r)] + \sigma_r \sqrt{h}\, \xi + O_{L^2}(h^{3/2}) \\ &= X_r + h\lambda^2 \left[u_r(X_r) - \frac{\dot{a}_r}{a_r} X_r\right] + \sigma_r \sqrt{h}\, \xi + O_{L^2}(h^{3/2}). \end{aligned} \tag{20}$$

This recovers both the reverse-step expansion in Equation (5) and the local policy in Equation (6). The correction is produced by composing the forward anchor transport with reverse resampling. The following proof bounds the remainder under a shared Brownian increment.

Theorem 3.3 (Error of the local-anchor approximation). Let $X_r = \psi_{t\to r}(X_t)$ and $X_{r+h} = \psi_{r\to r+h}(X_r)$ use exact transport. Let $\widetilde{X}_r^\star$ be the exact reverse-SDE sample from X_{r+h} to time r, and let $\widetilde{X}_r^{\mathrm{loc}}$ be Equation (6) with exact velocity. Couple both samples using the same Brownian increment, $\xi = W_h/\sqrt{h}$. Under the smoothness and moment assumptions in Appendix A.2,

$$\|\widetilde{X}_r^\star - \widetilde{X}_r^{\mathrm{loc}}\|_{L^2} = O(h^{3/2}) \qquad \text{as } h \to 0.$$

Write $\tau = r + h$ and $B_s(x) = u_s(x) - \frac{1}{2}\sigma_s^2 q_s(x)$ for the reverse-time drift. On an interval containing $[r, r+h]$, assume that a_s is bounded away from zero; u_s and B_s are globally Lipschitz in space, continuously differentiable in time, and have linear-growth bounds, including for their time derivatives; and σ_s is continuously differentiable with bounded derivative. Assume $\mathbb{E}\|X_r\|^2 < \infty$. These conditions make the following bounds uniform for sufficiently small h and hold for smooth coefficients with the corresponding bounded-derivative and moment bounds.

Proof. Parameterize reverse time by $v \in [0, h]$. The exact resampling process satisfies $\widehat{X}_0 = X_\tau$ and $d\widehat{X}_v = -B_{\tau-v}(\widehat{X}_v)\, dv + \sigma_{\tau-v}\, dW_v$, where W is independent of X_r. Thus $\widetilde{X}_r^\star = \widehat{X}_h$. Use the same increment $\xi = W_h/\sqrt{h}$ in the local approximation. Exact deterministic transport gives

$$X_\tau = X_r + h u_r(X_r) + R_{\mathrm{ode}}, \qquad \|R_{\mathrm{ode}}\|_{L^2} = O(h^2). \tag{21}$$

The reverse-SDE integral can be decomposed as

$$\begin{aligned} \widehat{X}_h &= X_\tau - h B_r(X_r) + \sigma_r W_h + R_b + R_\sigma, \\ R_b &= -\int_0^h \left[B_{\tau-v}(\widehat{X}_v) - B_r(X_r)\right] dv, \\ R_\sigma &= \int_0^h (\sigma_{\tau-v} - \sigma_r)\, dW_v. \end{aligned} \tag{22}$$

The moment and Lipschitz assumptions imply $\|\widehat{X}_v - X_\tau\|_{L^2} \le C\sqrt{v}$ and $\|X_\tau - X_r\|_{L^2} \le Ch$. Consequently, Minkowski's inequality gives

$$\|R_b\|_{L^2} \le C \int_0^h (h + \sqrt{v})\, dv = O(h^{3/2}). \tag{23}$$

By Itô isometry and the time regularity of σ, $\|R_\sigma\|_{L^2}^2 = d\int_0^h |\sigma_{\tau-v} - \sigma_r|^2 dv = O(h^3)$. Substituting the transport expansion and the score identity gives

$$\begin{aligned}\widehat{X}_h &= X_r + h\big[u_r(X_r) - B_r(X_r)\big] + \sigma_r\sqrt{h}\,\xi + R_{\mathrm{ode}} + R_b + R_\sigma \\ &= X_r + h\lambda^2\left[u_r(X_r) - \frac{\dot{a}_r}{a_r}X_r\right] + \sigma_r\sqrt{h}\,\xi + R_{\mathrm{ode}} + R_b + R_\sigma.\end{aligned} \tag{24}$$

The leading terms are precisely $\widetilde{X}_r^{\mathrm{loc}}$. The triangle inequality therefore yields the claimed $O(h^{3/2})$ strong error. This is a local discretization bound with exact transport and velocity; learned-model approximation errors are separate. □

Endpoint-anchor conditional and one-time implementation. For $\tau = 1$, the affine-path boundary conditions give $X_1 \sim p_1$ and $X_r \mid X_1 = x_1 \sim \mathcal{N}(a_r x_1, b_r^2 I)$. Exact transport produces $X_1 = \psi_{r\to 1}(\psi_{t\to r}(X_t)) \sim p_1$, so

$$\widetilde{X}_r = a_r X_1 + b_r \xi, \qquad \xi \sim \mathcal{N}(0, I), \quad \xi \perp X_1 \quad \Longrightarrow \quad \widetilde{X}_r \sim p_r. \tag{25}$$

This is the endpoint instance of Theorem 3.2 and yields Equation (8). It is exact for the affine conditional; it does not require a short-interval approximation. For accurate flow maps, the composition law gives $\psi_{r\to 1} \circ \psi_{t\to r} = \psi_{t\to 1}$. A one-time model therefore supplies the endpoint directly through $f_t^\theta(x_t)$, giving mean $a_r f_t^\theta(x_t)$ and variance $b_r^2 I$ as used in Section 3.3. This implementation choice does not restrict the general endpoint-anchor construction to one-time models. Appendix A.6 gives both sampling interfaces. At $r = 1$, $b_1 = 0$ and the final prediction is deterministic.

The path-preservation theorem uses exact transport and the exact conditional kernel. The local Gaussian rule approximates its reverse conditional with the error quantified above, whereas the endpoint Gaussian conditional is analytic and its model implementation depends on endpoint-prediction accuracy. Both provide explicit policy densities at stochastic steps with positive variance.

A.3 Gaussian likelihoods and optimization

At a stochastic transition, let $\pi_\theta(x' \mid s) = \mathcal{N}(\mu_\theta(s), \nu^2 I)$ with $\nu > 0$ fixed by the sampling schedule and $x' \in \mathbb{R}^d$. The joint log density and its mean gradient are

$$\log \pi_\theta(x' \mid s) = -\frac{\|x' - \mu_\theta(s)\|^2}{2\nu^2} - d\log\nu - \frac{d}{2}\log(2\pi), \qquad \nabla_\theta \log \pi_\theta = \frac{(D_\theta \mu_\theta)^\top (x' - \mu_\theta)}{\nu^2}. \tag{26}$$

The local mean is the corrected transport in Equation (6), with variance $\sigma_r^2 h$. The endpoint mean is the transported endpoint scaled by a_r in Equation (8), with variance b_r^2. Current and rollout policies use the same variance and are evaluated at the same stored action and state. Hence their density ratio simplifies to

$$\rho_{ij} = \exp\left(\frac{\|x_{r_j}^{(i)} - \mu_j^{\mathrm{old}}(s_j^{(i)})\|^2 - \|x_{r_j}^{(i)} - \mu_j^\theta(s_j^{(i)})\|^2}{2\nu_j^2}\right). \tag{27}$$

This is the ratio defined in Section 3.3. If a KL term is used, the Gaussian KL with the same covariance is $D_{\mathrm{KL}}(\pi_\theta \| \pi_{\mathrm{ref}}) = \|\mu_\theta - \mu_{\mathrm{ref}}\|^2/(2\nu^2)$.

The implementations average log densities over latent coordinates. Writing $\ell_\theta = d^{-1}\log\pi_\theta$, their normalized ratio is $\rho_{ij}^{\mathrm{norm}} = \exp(\ell_\theta - \ell_{\mathrm{old}}) = \rho_{ij}^{1/d}$. The numerical clipping threshold is applied to this normalized ratio; it is distinct from the joint-density ratio in Equation (27). This convention rescales log-ratio changes for high-dimensional image latents. The reported experiments use zero KL coefficient.

Advantages use the condition-specific reward mean and rollout-batch standard deviation, followed by clipping, as defined in Section 3.3. ImageNet uses the current class-group mean, while text-to-image

training uses running per-prompt statistics (Appendix C.2). Rewards and sampled actions are held fixed during each update. Gradients are stopped through the auxiliary equal-time velocity in the local correction; the transport prediction supplies the trainable mean. Only LoRA parameters are updated. The deterministic terminal step produces the reward image and contributes no Gaussian likelihood term because its variance is zero.

A.4 Midpoint anchors and the posterior constraint

Proposition 3.4 (Posterior constraint for midpoint anchors). Let $X_s = a_s X_1 + b_s X_0$, where $X_0 \sim \mathcal{N}(0, I)$ is independent of X_1. For $r < \tau < 1$ with $a_\tau \neq 0$, define $c_{r,\tau} = b_r - a_r b_\tau / a_\tau$. Then

$$\frac{a_r}{a_\tau} x + c_{r,\tau} X_0^{\mathrm{post}} \sim p_{r|\tau}(\cdot \mid x), \qquad X_0^{\mathrm{post}} \sim p_{0|\tau}(\cdot \mid x).$$

If $c_{r,\tau} \neq 0$, replacing X_0^{post} with independent standard Gaussian noise is exact if and only if $p_{0|\tau}(\cdot \mid x) = p_0$. For the linear path and data with finite second moment, this substitution fails to recover the conditional kernel for every $0 \leq r < \tau < 1$. (see Appendix A.4 for proof.)

Proof. Consider a midpoint anchor $r < \tau < 1$ under the affine interpolant $X_s = a_s X_1 + b_s X_0$, with independent $X_0 \sim \mathcal{N}(0, I)$ and $X_1 \sim p_1$. The arithmetic midpoint $\tau = (r+1)/2$ is one such choice. For $a_\tau \neq 0$, eliminating X_1 gives

$$X_r = \frac{a_r}{a_\tau} X_\tau + \left(b_r - \frac{a_r b_\tau}{a_\tau} \right) X_0. \tag{28}$$

Although this identity is affine in X_0, conditioning on $X_\tau = x_\tau$ changes the distribution of that noise. When the data density exists, Bayes' rule gives

$$p_{0|\tau}(x_0 \mid x_\tau) \propto p_0(x_0)\, p_1\left(\frac{x_\tau - b_\tau x_0}{a_\tau} \right), \tag{29}$$

where the Jacobian is constant in x_0. The interpolant conditional is therefore sampled by

$$\frac{a_r}{a_\tau} x_\tau + \left(b_r - \frac{a_r b_\tau}{a_\tau} \right) X_0^{\mathrm{post}} \sim p_{r|\tau}(\cdot \mid x_\tau), \qquad X_0^{\mathrm{post}} \sim p_{0|\tau}(\cdot \mid x_\tau). \tag{30}$$

When $c_{r,\tau} \neq 0$, the affine map from X_0^{post} to X_r is invertible. Consequently, replacing the posterior law of X_0^{post} with p_0 yields the same conditional distribution exactly when the two noise laws agree. For the linear path, $c_{r,\tau} = (\tau - r)/\tau > 0$. Moreover, independence of X_0 and X_1 and the finite-second-moment assumption imply

$$\mathrm{Cov}(X_0, X_\tau) = (1 - \tau) I \neq 0. \tag{31}$$

Thus X_0 and X_τ are dependent, so $p_{0|\tau}(\cdot \mid x)$ cannot equal p_0 for almost every anchor state. The unconditional Gaussian substitution therefore differs from the true conditional on a set of anchor states with positive probability. This proves the claim for every $0 \leq r < \tau < 1$.

Replacing X_0^{post} with independent standard Gaussian noise generally does not implement this conditional, since a midpoint anchor contains information about both X_1 and X_0. At the endpoint, $a_1 = 1$ and $b_1 = 0$, so independence recovers the Gaussian kernel $a_r x_1 + b_r \xi$. Thus, the obstacle for midpoint anchors is the unavailable posterior, not the choice of a midpoint anchor itself. This affine-interpolant conditional and the reverse-diffusion construction used for the local policy are different couplings; the identity above does not equate their transition laws.

□

Midpoint-anchor experiment. We evaluate an OCR-trained MeanFlow adapter using a midpoint-anchor kernel with anchor fraction 0.5 and scheduled Gaussian re-noising. We follow the MeanFlow OCR training settings and use the same task test sets, resolution, and guidance scale as in Appendix C.2, replacing the local-anchor kernel with the midpoint-anchor kernel. We evaluate the EMA adapter after 1,000 optimizer steps using deterministic MeanFlow sampling across inference step counts. The adapter is evaluated with batch size two, as in its training-time evaluation; the reused Base results use batch size four.

At $K = 4$, midpoint-anchor post-training raises OCR only from 0.3491 to 0.3718, compared with 0.8504 for the local-anchor adapter evaluated at training step 800 (Table 9). At $K = 3$, OCR falls below Base. All five DrawBench metrics decrease at every reported number of sampling steps (Table 5). These results complement the controlled transition ablations in Section 4.1: tractable Gaussian noise alone does not ensure useful post-training. The weak task adaptation and broad quality degradation are consistent with the posterior mismatch analyzed above; the sweep measures its downstream consequences under deterministic evaluation, rather than directly measuring stochastic rollout marginals.

Table 5: MeanFlow OCR post-training with the midpoint-anchor kernel, evaluated using deterministic sampling and the EMA adapter at training step 1,000. Task Metrics measure RL improvements on the target tasks; DrawBench Metrics are auxiliary general-quality metrics used to monitor degradation outside the optimized reward. Bold indicates the better score within each Base/adapter pair. DB Sum is the unnormalized sum of the five DrawBench metrics. Unmeasured tasks are denoted by –.

K Model	Task Metrics			DrawBench Metrics					
	GenEval↑	OCR Acc.↑	PickScore Task↑	DB-PickScore↑	Aesthetic↑	DeQA↑	ImgRwd↑	UniRwd↑	DB Sum↑
3 MeanFlow Base	–	**0.2876**	–	**21.3577**	**5.4391**	**4.0001**	**0.2157**	**2.5495**	**33.5621**
3 MeanFlow + Midpoint GRPO LoRA	–	0.2767	–	21.0636	5.2149	3.3155	-0.0996	2.2472	31.7417
4 MeanFlow Base	–	0.3491	–	**21.6298**	**5.5368**	**4.1712**	**0.3918**	**2.7187**	**34.4483**
4 MeanFlow + Midpoint GRPO LoRA	–	**0.3718**	–	21.4442	5.3572	3.7296	0.1650	2.4845	33.1805
7 MeanFlow Base	–	0.4517	–	**22.0761**	**5.6090**	**4.2500**	**0.5744**	**2.9216**	**35.4311**
7 MeanFlow + Midpoint GRPO LoRA	–	**0.5020**	–	21.9719	5.4718	4.0544	0.4473	2.8023	34.7476
15 MeanFlow Base	–	0.5507	–	**22.2302**	**5.6489**	**4.3217**	**0.6870**	**3.0208**	**35.9086**
15 MeanFlow + Midpoint GRPO LoRA	–	**0.5865**	–	22.1645	5.5345	4.2350	0.5891	2.9602	35.4833

Stochastic rollout quality. Figure 6 shows five-step samples from the midpoint-anchor policy during late-stage training. The visible grain, distorted textures, and malformed lettering show that poor sample quality is already present in the trajectories used to collect rewards. Together with the posterior constraint above and the weak downstream results in Table 5, these observations support the importance of preserving the intended marginals when constructing the exploration policy.



Figure 6: Five-step stochastic training rollouts from the MeanFlow midpoint-anchor policy. These late-training samples exhibit grain, texture artifacts, and malformed text under the policy used for reward collection. The deterministic evaluation sweep is reported separately in Table 5.

A.5 Unified Interpretation of Anchor Choices

The local-anchor policy can also be viewed as a Langevin-style corrector induced by stochastic flow-map composition: nearby-anchor transport followed by a short stochastic reverse-resampling step, including its drift component, yields a correction with matched drift and noise around the deterministic flow-map output (Equation (6)). The endpoint-anchor policy, in contrast, recovers the familiar clean-prediction-and-re-noising structure of consistency samplers (Equation (8)). These two cases correspond to anchor choices for which the required reverse conditional is either available in closed form or admits a controlled local approximation (Theorem 3.3). A generic midpoint anchor does not enjoy the same tractability: its conditional depends on a nontrivial posterior over the original noise variable, so the independent Gaussian re-noising rule considered here is generally mismatched (Proposition 3.4). This does not preclude midpoint anchors in principle; they could be realized with an appropriate posterior or conditional sampler, or with an approximation sufficiently accurate for the desired transition.

More broadly, these cases illustrate our view of stochasticizing deterministic flow maps. Rather than perturbing the output of a long-range map with an arbitrary noise kernel, we construct stochastic transitions by composing deterministic transport with conditional resampling through an anchor. The local Langevin-style corrector and the consistency-model re-noising sampler then arise as two tractable instances of this same principle. From this perspective, the anchor determines both where stochasticity is introduced and the conditional structure required to preserve the underlying probability path. For the local anchor, this structure includes the pairing of drift and noise. This view explains both the failure of naive midpoint re-noising and the strong degradation observed when the local drift correction is removed (Table 5 and Section 4.1).

A.6 Evaluation Sampling for One-Time and Two-Time Models

A two-time model directly predicts $\psi^{\theta}_{t\to r}(x_t)$; MeanFlow implements it through an average velocity. A one-time model directly predicts only $f^{\theta}_t(x_t) = \psi^{\theta}_{t\to 1}(x_t)$; an intermediate sample is constructed by the chosen sampler from that endpoint estimate. For the affine stochastic sampler, this is $x_r = a_r f^{\theta}_t(x_t) + b_r \xi$. Algorithm 2 describes sampling during evaluation; stochastic rollout collection for RL post-training follows Algorithm 1.

Algorithm 2 Evaluation sampling with two-time maps or one-time endpoint predictors

Require: Grid $0 = r_0 < \cdots < r_{K+1} = 1$, condition c, generator, path (a_r, b_r)
Sample $x_{r_0} \sim p_0$
for $j = 1, \ldots, K+1$ **do**
  if the generator directly predicts two-time maps **then**
    $x_{r_j} \leftarrow \psi^{\theta}_{r_{j-1}\to r_j}(x_{r_{j-1}} \mid c)$
  else
    $\widehat{x}_1 \leftarrow f^{\theta}_{r_{j-1}}(x_{r_{j-1}} \mid c)$
    if $j \le K$ **then**
      Sample $\xi_j \sim \mathcal{N}(0, I)$ independently
      $x_{r_j} \leftarrow a_{r_j}\widehat{x}_1 + b_{r_j}\xi_j$
    else
      $x_1 \leftarrow \widehat{x}_1$
    end if
  end if
end for
return x_1

B Extended Related Work

Continuous-time generative models. Diffusion models and continuous-time flow-based models provide a foundation for high-quality generative modeling. Diffusion models define a stochastic noising process and learn to reverse it through denoising or score estimation (Ho et al., 2020; Song et al., 2020a;b). Flow matching and rectified flow learn time-dependent velocity fields that transport a

simple prior distribution to the data distribution through an ODE (Lipman et al., 2022; Liu et al., 2023). Stochastic interpolants connect deterministic flows and stochastic diffusions through probability paths (Albergo et al., 2025). These formulations support principled generative sampling, but their standard numerical solvers typically require many sequential evaluations. Our setting concerns generators that have already replaced this integration with learned long-range maps.

Few-step flow-map generation. Few-step methods reduce sampling cost by learning long-range transport operators. Consistency models learn mappings from noisy states to clean data and support one-step or few-step generation through self-consistency constraints (Song et al., 2023; Song & Dhariwal, 2024; Geng et al., 2024). Flow Map Matching provides a framework for learning two-time flow maps of probability-flow dynamics and connects these maps with consistency models and progressive distillation (Boffi et al., 2025). Shortcut models and MeanFlow parameterize long-range transitions or average velocities, enabling direct jumps across time intervals (Frans et al., 2025; Geng et al., 2025a;b). The learned map $\psi_{t\to r}$ is deterministic, even when a sampling procedure composes it with re-noising. A deterministic map alone does not define a nondegenerate stochastic policy or the transition-density ratios needed by GRPO. Flow-Map GRPO supplies this policy interface while preserving the pretrained prediction parameterization.

Reinforcement learning for generative models. Reinforcement learning aligns generative models with task-level rewards, including human preference, aesthetic quality, and downstream evaluation metrics. DDPO formulates diffusion sampling as a multi-step Markov decision process and applies policy-gradient optimization to stochastic denoising trajectories (Black et al., 2024). DanceGRPO, Flow-GRPO, and MixGRPO develop related approaches for visual generation and flow-based models (Xue et al., 2025; Liu et al., 2026; Li et al., 2025). In particular, Flow-GRPO introduces stochasticity into velocity-based flow models through an SDE formulation, obtaining stochastic transitions and likelihood ratios for GRPO. GLASS Flows studies stochastic transition sampling for inference-time alignment of continuous-time flow and diffusion models (Holderrieth et al., 2026b).

The distinction is the generator's transition interface. Velocity-based samplers admit local stochastic updates obtained by discretizing their continuous-time dynamics. A few-step flow-map generator instead predicts a long-range transport directly. Over such an interval, an SDE transition integrates the effects of the intervening stochastic path and is generally non-Gaussian; its conditional law is not determined by an independent Gaussian perturbation of the deterministic endpoint. Thus, applying a local velocity-based stochasticization rule to a long-range map requires an additional policy construction. Our anchored composition separates deterministic transport from stochastic resampling and provides explicit transition densities for likelihood-ratio optimization.

Reward-based post-training of consistency models. RLCM (Oertell et al., 2024) formulates the multistep consistency sampler as a Gaussian policy and applies clipped policy-gradient updates to terminal image rewards. Its clean prediction followed by Gaussian re-noising is closely related to our endpoint-anchor case. ROCM (Shekhar & Zhang, 2025) instead optimizes consistency models using direct reward gradients with distributional regularization. These works establish that consistency-specific reward optimization predates the concurrent methods below. Our contribution extends beyond this endpoint setting: ASFMC supplies a common construction for endpoint predictors and two-time transport maps, including a local-anchor policy for MeanFlow and an analysis of the posterior constraint on intermediate anchors.

Stochastic and reward-aware flow maps. Several methods learn stochastic variants of flow maps for posterior sampling and reward alignment. Meta Flow Maps extend consistency models and flow maps to one-step posterior sampling, producing different samples from $p(x_1 \mid x_t)$ for a fixed intermediate state (Potaptchik et al., 2026). Diamond Maps develops stochastic flow-map models designed for efficient reward alignment (Holderrieth et al., 2026a). Strong Stochastic Flow Maps learns strong solution maps of additive-noise SDEs (McCallum et al., 2026), while stochastic few-step models study efficient sampling from SDE-defined conditional distributions (Passaro et al., 2026). Flow Map Reward Guidance uses flow maps for training-free reward guidance from an optimal-control perspective (Huang et al., 2026a).

These works address several distinct settings: learning native stochastic maps, approximating posterior or SDE solution operators, and guiding generation at inference time. We address RL post-training

of an existing deterministic flow-map generator. **ASFMC preserves the deterministic flow-map parameterization.** It composes the pretrained transport with anchor-based resampling to obtain a stochastic policy, and reward optimization updates adapters through GRPO. The local and endpoint constructions support two-time and one-time predictors within the same framework. For the local policy, the continuous-time derivation motivates the coupled drift and noise terms; we do not assume that a finite-strength discretization exactly reproduces the underlying SDE transition.

Concurrent work. Concurrent to our work, MeanFlowNFT (Huang et al., 2026b) brings DiffusionNFT's forward-process RL objective to average-velocity generators. Its central issue is the mismatch between the instantaneous velocity optimized by DiffusionNFT and the average velocity used by MeanFlow. It uses the MeanFlow identity to construct an induced instantaneous-velocity predictor for training, while retaining average-velocity sampling. RAVEN (Lu et al., 2026) introduces CM-GRPO for video post-training, defining GRPO on the Gaussian transition induced by a consistency sampler's clean-endpoint prediction. We introduce a unified framework for likelihood-ratio RL with flow-map generators. Anchored stochastic composition resolves the policy-construction problem through local and endpoint resampling, covering both MeanFlow-style two-time maps and consistency-style one-time predictors without changing their prediction interfaces.

Positioning. Our contribution is a unified stochastic policy construction for pretrained few-step flow-map generators. ASFMC provides tractable exploration and transition densities while retaining the pretrained model's architecture and parameterization. This enables reward-based adaptation of MeanFlow and consistency models through a common GRPO objective. It also separates the design of the stochastic training policy from the generator's inference sampler, allowing the adapted model to retain its one-step or few-step generation interface.

C Experiment Details

C.1 MeanFlow and CM Ablation Details

Base model and reward. We use a pretrained class-conditional MeanFlow SiT-B/4 checkpoint with distilled classifier-free guidance at 256×256 resolution. The generator operates in a $4 \times 32 \times 32$ VAE latent space. The base weights are frozen, and LoRA adapters with rank 64 and scaling factor 128 are applied to the attention and feed-forward projections. For each class, the reward compares a group of 24 generated images with 50 real reference images using the negative unbiased MMD-squared estimate of Miao et al. (2024). We use 2,048-dimensional InceptionV3 features (Szegedy et al., 2016) and the polynomial kernel $k(z, z') = (z^\top z'/2048 + 1)^3$. The reward for image x_m is its leave-one-out contribution,

$$\hat{R}(x_m) = R_{\mathrm{MMD}}(\mathbb{X}_g; \mathbb{X}_r) - R_{\mathrm{MMD}}(\mathbb{X}_g \setminus \{x_m\}; \mathbb{X}_r), \tag{32}$$

where $\mathbb{X}_g$ and $\mathbb{X}_r$ denote the generated and reference sets. Thus, the GRPO group also serves as the image set used to compute the reward.

Local-anchor ablation study: implementation. For clarity, we describe the implementation using reverse time $s = 1 - t$, where $s = 1$ denotes noise and $s = 0$ denotes data, opposite to the time convention in the main text. We construct the five-step rollout grid by taking five uniformly spaced nodes from 1 to 0.05 and appending the data endpoint 0, giving $(1, 0.7625, 0.525, 0.2875, 0.05, 0)$. For a stochastic step $s_i \to s_{i+1}$, let $h_i = 0.01(s_i - s_{i+1})$ and denote the deterministic transition by $\psi^\theta_{s_i \to s_{i+1}}(x_{s_i})$. The full transition specializes Equation (6) to the linear path, with this local anchor interval:

$$\begin{aligned} d_i &= -\frac{h_i \lambda^2}{1 - s_{i+1}} \left(\psi^\theta_{s_i \to s_{i+1}}(x_{s_i}) + (1 - s_{i+1})\, \mathrm{sg}[u_\theta(\psi^\theta_{s_i \to s_{i+1}}(x_{s_i}), s_{i+1})] \right), \\ x_{s_{i+1}} &= \underbrace{\psi^\theta_{s_i \to s_{i+1}}(x_{s_i})}_{\text{transport}} + \underbrace{d_i}_{\text{drift correction}} + \underbrace{\sigma_i \xi_i}_{\text{resampling noise}}, \\ \sigma_i &= \lambda \sqrt{\frac{2 h_i s_{i+1}}{1 - s_{i+1}}}, \qquad \xi_i \sim \mathcal{N}(0, I), \end{aligned} \tag{33}$$

where u_θ is the instantaneous velocity, evaluated by setting the MeanFlow time interval to zero, and sg stops gradients through this velocity prediction. We set $\lambda = 5$ for ImageNet. The four stochastic steps have standard deviations approximately $(0.617, 0.362, 0.219, 0.079)$. The *No drift* ablation sets $d_i = 0$ and keeps these standard deviations. Both constant-noise variants set $\sigma_i = \bar{\sigma}$, where $\bar{\sigma}^2 = \frac{1}{4}\sum_{i=0}^{3} \sigma_i^2$, giving $\bar{\sigma} \approx 0.376$. The four table rows therefore use (d_i, σ_i) (Ours), $(d_i, \bar{\sigma})$ (Drift + const. noise), $(0, \sigma_i)$ (No drift), and $(0, \bar{\sigma})$ (No drift + const. noise). Each step uses independent Gaussian noise. The final transition from 0.05 to 0 is deterministic and excluded from the policy objective in all variants.

The no-drift variants therefore implement the most direct stochasticization of MeanFlow: first compute its deterministic long-range output, then add Gaussian noise. Explicitly,

$$\begin{aligned} \psi^{\theta}_{s_i \to s_{i+1}}(x_{s_i}) &= x_{s_i} - (s_i - s_{i+1})\bar{u}^{\theta}_{s_i \to s_{i+1}}(x_{s_i}), \\ x_{s_{i+1}} &= \psi^{\theta}_{s_i \to s_{i+1}}(x_{s_i}) + \eta_i \xi_i, \qquad \eta_i \in \{\sigma_i, \bar{\sigma}\}. \end{aligned} \tag{34}$$

Here $\bar{u}^\theta$ is the predicted average velocity, and $\eta_i = \sigma_i$ and $\eta_i = \bar{\sigma}$ correspond to *No drift* and *No drift + const. noise*, respectively. Noise is added to the transported state $\psi^{\theta}_{s_i \to s_{i+1}}(x_{s_i})$, leaving the deterministic MeanFlow update unchanged. These baselines isolate whether output noise alone suffices for RL post-training; our local-anchor policy additionally supplies the drift correction d_i paired with its noise schedule.

Endpoint-anchor ablation study: implementation. For the ImageNet CM implementation, we reuse the pretrained MeanFlow SiT-B/4 weights with the endpoint predictor $f_\theta(x_s, s) = x_s - s\bar{u}^{\theta}_{s\to 0}(x_s)$, where $\bar{u}^\theta$ denotes the MeanFlow average velocity in the reverse-time convention. The target time is fixed to zero for every network evaluation. On the same rollout grid, our endpoint-anchor policy (*Ours*) in Equation (8) becomes

$$x_{s_{i+1}} = \underbrace{(1 - s_{i+1}) f_\theta(x_{s_i}, s_i)}_{\text{scaled endpoint prediction}} + \underbrace{s_{i+1}\xi_i}_{\text{resampling noise}}, \qquad \xi_i \sim \mathcal{N}(0, I). \tag{35}$$

The noise standard deviations of *Ours* are $(0.7625, 0.525, 0.2875, 0.05)$. The constant-noise variant keeps the mean unchanged and replaces these scales with $\bar{\sigma}_{\mathrm{CM}} = (\frac{1}{4}\sum_{i=0}^{3} s_{i+1}^2)^{1/2} \approx 0.485$. Only the noise coefficient changes: the endpoint predictor and its multiplier $1 - s_{i+1}$ remain fixed, and the summed variance over the four stochastic steps is preserved. The last endpoint prediction is deterministic.

Training setting. Each epoch samples 48 class groups with 24 images per group and performs two optimizer updates. We use AdamW with learning rate 3×10^{-4}, weight decay 10^{-4}, and gradient clipping at 1.0. Advantages are centered within each group, divided by the global reward standard deviation, and clipped to $[-5, 5]$. The GRPO clip range is 10^{-4} and the KL coefficient is zero. Transition log probabilities are averaged over latent dimensions. Training uses fp32 on a single H100 GPU, without a time shift. All variants are trained for 400 epochs under the same configuration, and we evaluate the final checkpoints. All reported runs use training seed 42.

Evaluation setting. Following the evaluation setting of Miao et al. (2024), we compare generated and held-out real-image sets using FID, Precision, and Recall. We use ImageNet classes 0–99. For each class, the first 50 training images form the reward reference set, the next 20 form the held-out test set, and the next 30 form a separate validation set. These sets do not overlap. Evaluation compares 50 generated images per class (5,000 in total) with 20 test images per class (2,000 in total). FID, Precision, and Recall are computed from pooled InceptionV3 features (Szegedy et al., 2016) across all 100 classes using the same PyTorch evaluator for every row in Table 1. The main table uses deterministic five-step MeanFlow sampling for the MeanFlow-trained adapters and native five-step CM sampling for the CM-trained adapters. The native CM sampler uses the noise coefficients in Equation (35) and a fixed evaluation seed, including for the constant-noise-trained adapter; this evaluation is stochastic. Each sampler is compared with its own pretrained baseline. One-step evaluation uses the same direct noise-to-data transition for both policies.

C.2 Single-Score and Mixed-Score Post-Training Details

Base checkpoints. We use the FLUX.1-lite few-step generators released by T2I-Distill Pu et al. (2025). The released checkpoints include both MeanFlow and sCM variants. MeanFlow is treated as a two-time flow-map model, which directly parameterizes transitions $\psi^{\theta}_{t\to r}$, while sCM is treated as a one-time endpoint flow-map model, which predicts $\psi^{\theta}_{t\to 1}$. For each method, the "Base" model denotes the corresponding released T2I-Distill checkpoint without any RL post-training. The post-trained model denotes the same frozen base generator plus a reward-specific LoRA adapter trained with Flow-Map GRPO.

Reward-specific post-training. We train separate LoRA adapters for four reward objectives:

$$R \in \{\mathrm{OCR},\ \mathrm{PickScore},\ \mathrm{GenEval},\ R_{\mathrm{mix}}\}. \tag{36}$$

Single-reward adapters use the corresponding training prompt set; mixed-reward adapters sample OCR and PickScore prompts in equal proportions. In each case, the model is post-trained on these prompts and the reward is evaluated on the final generated image Liu et al. (2026). The runs share the policy objective and LoRA architecture; Table 7 lists their training steps. In particular, the pretrained generator is frozen and only the LoRA adapter is updated. This setup allows us to evaluate whether Flow-Map GRPO can improve few-step flow-map generators without full-parameter finetuning.

Flow-Map GRPO rollout. During RL post-training, we use $K = 4$ stochastic flow-map transitions for all models and all rewards. The rollout has six time nodes defining four stochastic policy transitions and a final deterministic endpoint transition. The stochastic transitions are used to compute policy likelihood ratios, while the final deterministic transition is used only to obtain the decoded image for reward evaluation. For a prompt c, we sample a group of G trajectories, compute the final-image rewards, normalize the rewards into group-level advantages, and optimize the LoRA policy with the clipped GRPO objective in Equation 10.

Common training hyperparameters. The following settings describe OCR, GenEval, and mixed-reward post-training. Per-device batch sizes depend on the experiment.

Table 6: Common Flow-Map GRPO post-training hyperparameters for text-to-image experiments.

Hyperparameter	Value
Base model	T2I-Distill FLUX.1-lite MeanFlow / sCM checkpoints
Trainable parameters	LoRA only
Frozen parameters	Pretrained flow-map generator
Resolution	512×512
Guidance scale	3.5
Max text sequence length	256
Stochastic policy steps	$K = 4$
Images per prompt / group size	$G = 24$
Training batch size	Experiment-dependent
Number of GPUs	8 H100 GPUs
Mixed precision	bf16
Gradient accumulation steps	24
Optimizer	AdamW
Learning rate	3×10^{-4}
Adam betas	$(0.9, 0.999)$
Weight decay	10^{-4}
Max gradient norm	1.0
GRPO clip range	10^{-4}
Advantage clipping	5
KL coefficient β	0.0
EMA	enabled
EMA decay	0.9
EMA update interval	8 optimizer steps
Seed	42

LoRA configuration. All adapters use rank 64, scaling factor 128, and zero dropout. Following Liu et al. (2026), LoRA is applied to the attention query, key, value, and output projections, including the additional conditioning projections, and to both feed-forward projections in the image and context streams.

Advantage normalization and GRPO objective. For each prompt group, rewards are gathered across devices and normalized before the policy update. We use per-prompt reward statistics together with a global batch standard deviation. Concretely, for a trajectory i associated with prompt c, the advantage is computed as

$$A_i = \frac{R(x_1^i|c) - \mu_c}{\sigma_{\text{global}} + \epsilon_{\text{std}}}, \tag{37}$$

where μ_c is the running per-prompt reward mean, σ_{global} is the reward standard deviation computed over the gathered rollout batch, and $\epsilon_{\text{std}} = 10^{-4}$ is a numerical stabilizer. We then clip $\widehat{A}_i = \text{clip}(A_i, -5, 5)$. The policy update recomputes the log probability of each stochastic ASFMC transition and applies the clipped GRPO objective.

Stochasticization during training. MeanFlow uses the local-anchor Gaussian transition in Equation (6), with reverse-time coordinates $s = 1 - t$. Its average-velocity prediction $\bar{u}^\theta$ gives the deterministic transport $\psi^\theta_{s_i \to s_{i+1}}(x_{s_i}) = x_{s_i} - (s_i - s_{i+1})\bar{u}^\theta_{s_i \to s_{i+1}}(x_{s_i})$. The instantaneous velocity is obtained by evaluating the same network with zero time interval, $u_\theta(x, s) = \bar{u}^\theta_{s \to s}(x)$. We evaluate this velocity at $x = \psi^\theta_{s_i \to s_{i+1}}(x_{s_i})$ and $s = s_{i+1}$, stop gradients through the velocity prediction, and apply the drift correction and Gaussian noise. We use $\lambda = 7$ and $h_i = 0.01(s_i - s_{i+1})$.

We take five uniformly spaced base nodes from 1 to 0.05, append 0, and apply the timestep transformation used in T2I-Distill (Pu et al., 2025) to obtain the rollout times s_i, from which we compute h_i. The terminal base-grid node is 0.05, and the injected noise standard deviation is $\lambda\sqrt{2h_i s_{i+1}/(1 - s_{i+1})}$. The final endpoint segment is deterministic.

For sCM, $f_t^\theta(x_t)$ denotes its learned endpoint predictor, which takes the noisy latent x_t and time t as input and directly predicts the clean-image latent at the fixed data endpoint $t = 1$. We use this prediction as the endpoint anchor, $X_1 = f_t^\theta(x_t) = \psi^\theta_{t \to 1}(x_t)$, in the policy of section 3.2. The resulting transition is $\widetilde{X}_r = a_r f_t^\theta(x_t) + b_r \xi$, with $\xi \sim \mathcal{N}(0, I)$, and requires one network call per stochastic step. For both backbones, only stochastic transitions contribute to the GRPO likelihood-ratio objective.

Reward mixing. Mixed-reward training samples OCR and PickScore prompts with equal probability. OCR prompts receive both rewards, whereas PickScore prompts receive only PickScore. We normalize each component separately within each reward-availability subset using pooled within-group normalization. For subset a, let I_a contain its rollout samples, let $\bar{R}_{q,a}$ be the subset mean for component q, and let $\bar{R}_{q,g(i)}$ be its mean in prompt group $g(i)$. We compute

$$s^2_{q,a} = \frac{1}{|I_a|} \sum_{i \in I_a} (R_{q,i} - \bar{R}_{q,g(i)})^2, \qquad \widetilde{R}_{q,i} = \frac{R_{q,i} - \bar{R}_{q,a}}{s_{q,a} + 10^{-6}}. \tag{38}$$

A component with scale at most 10^{-6} contributes zero. Available weights are renormalized to sum to one: $R_{\text{mix}} = 0.7\widetilde{R}_{\text{OCR}} + 0.3\widetilde{R}_{\text{Pick}}$ applies to OCR prompts, and $R_{\text{mix}} = \widetilde{R}_{\text{Pick}}$ to PickScore prompts. We then apply the standard prompt-relative advantage normalization. The OCR/PickScore weights remain fixed at 0.7/0.3. For GenEval, we follow the standard evaluation procedure of Ghosh et al. (2023) and use the resulting compositional score as the reward.

Training steps. Table 7 lists the training steps and hardware for the text-to-image experiments. Results use EMA adapters and one training seed per configuration.

Table 7: Training steps for the evaluated text-to-image EMA adapters. GPU counts refer to training; offline evaluations use one H100.

Backbone	Reward	Checkpoint step	H100s
MeanFlow	OCR	800	8
sCM	OCR	700	8
MeanFlow	PickScore	1,500	8
sCM	PickScore	1,500	8
MeanFlow	GenEval	1,000	8
sCM	GenEval	1,000	8
MeanFlow	OCR 0.7 + PickScore 0.3	1,000	8
sCM	OCR 0.7 + PickScore 0.3	1,000	8

Evaluation setting. The sweeps are evaluated on one H100, at 512×512 resolution, guidance 3.5, and bf16 precision. Base and LoRA use matched prompts, sampling schedules, and random-number initialization within each family. MeanFlow disables post-flow resampling noise and uses per-sample seeds $42 + i$. sCM OCR, PickScore, and DrawBench use the native stochastic sampler with global CUDA RNG seeded at 42 and $\sigma_{\text{data}} = 0.5$. sCM GenEval uses per-sample seeds. The tables use $K + 1$ configured sampler steps. MeanFlow reports $K \in \{3, 4, 7, 15\}$ and sCM reports $K \in \{2, 3, 4, 7, 15\}$.

Following the OCR scoring implementation of Flow-GRPO (Liu et al., 2026), we evaluate 1,018 test prompts using PaddleOCR (Du et al., 2020). Its score is one minus normalized, clipped Levenshtein distance to the first quoted string, after lowercasing and removing spaces; a substring match receives full credit. PickScore Task uses 2,048 separate preference-test prompts and raw PickScore logits (Kirstain et al., 2023). Following the standard GenEval evaluation (Ghosh et al., 2023), we use 2,212 images (553 prompts, four samples each), the Mask2Former/CLIP scorer, and the equal-weight mean of six category accuracies.

We use the 200 prompts from DrawBench (Saharia et al., 2022), generating five images per prompt (1,000 images). We evaluate PickScore (Kirstain et al., 2023), the LAION aesthetic predictor (Schuhmann, 2022), DeQA (You et al., 2025), ImageReward (Xu et al., 2023), and UnifiedReward (Wang et al., 2025) on these images. Both backbones use raw 1–5 DeQA and UniReward scores. DB Sum is the unweighted sum of the five DrawBench metrics.

C.3 Anchor Choice for MeanFlow Text-to-Image Post-Training

To assess anchor choice in our text-to-image setting, we compare deterministic sampling, our local-anchor policy, and the endpoint-anchor policy using the same pretrained FLUX.1-lite MeanFlow checkpoint before RL post-training. All three use five sampling steps ($K = 4$), matched prompts and seeds, and the evaluation settings in Appendix C.2.

Table 8: Five-step sampling with the pretrained FLUX.1-lite MeanFlow checkpoint in our text-to-image setting, before RL post-training. Metric definitions follow Appendix C.2; DB Sum includes all five DrawBench metrics, including those not individually displayed.

Sampler	**Task Metrics**		**DrawBench Metrics**		
	OCR↑	**PickScore↑**	**DeQA↑**	**ImgRwd↑**	**DB Sum↑**
Deterministic (standard evaluation)	0.349	20.98	4.17	0.39	34.45
Local anchor (Ours)	0.444	20.72	3.03	0.38	32.68
Endpoint anchor (full noise)	0.208	17.51	2.68	−1.80	25.70

Under this text-to-image setting, Table 8 shows lower scores for endpoint-anchor sampling than for deterministic sampling and our local-anchor policy on every reported metric. These results motivate our choice of the local anchor for T2I post-training of this FLUX.1-lite MeanFlow checkpoint.

C.4 Baseline Setting and Challenges Posed by Flow Maps

Table 3 and Figure 7 compare SFT and DiffusionNFT, representing two widely used classes of post-training methods, with Flow-Map GRPO under matched training and evaluation settings. The central challenge for applying these post-training methods to flow maps arises from the prediction quantities learned by flow maps: a long-range transport or endpoint map is generally neither an instantaneous velocity nor a posterior-mean denoiser. Native flow-map supervision also couples these predictions to an underlying instantaneous field through transport or consistency identities. Consequently, reward-selected endpoints can conflict with a frozen teacher's dynamics, while forward-noising regression can target a different quantity from the learned flow map. These are objective mismatches rooted in flow-map structure; matching optimization settings alone does not resolve them. This highlights the value of Flow-Map GRPO, which addresses the challenges posed by flow maps in RL post-training.

Matched comparison settings. We control the shared experimental settings within each backbone and task: the pretrained checkpoint, training and test prompt splits, reward function, LoRA-only adaptation, 512×512 resolution, guidance scale 3.5, text length 256, five-step collection and evaluation, and EMA evaluation. The recipes use AdamW with learning rate 3×10^{-4}, 24 generated images per prompt, and gradient accumulation of 24. Table 3 matches the optimizer-step endpoint across methods within each column: 700 for sCM OCR, 800 for MeanFlow OCR, and 1,500 for both PickScore comparisons. The comparison uses the same number of optimizer steps. Evaluation settings are specified in Appendix C.2.

Online SFT objective. For each condition c, let $\{(x_1^{(i)}, R_i)\}_{i=1}^{G}$ denote the collected clean latents (at the main text's data endpoint $t = 1$) and their rewards. Let $F_\theta^{(i)}$ and $F_{\text{base}}^{(i)}$ denote the current and frozen-base backbone predictions at the same noisy input, conditioning, and sampled times, with these arguments suppressed. Define the per-sample reference penalty as $\ell_{\text{ref},i} = d^{-1}\|F_\theta^{(i)} - \text{sg}[F_{\text{base}}^{(i)}]\|^2$, where d is the number of prediction elements. Following Flow-GRPO (Liu et al., 2026), online SFT keeps all samples tied for the highest reward within each condition and minimizes

$$m_i = \mathbf{1}\left\{R_i = \max_{1\le k\le G} R_k\right\}, \qquad \mathcal{L}_{\text{SFT}} = \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\left(m_i \ell_{\text{native},i} + \alpha_{\text{SFT}}\ell_{\text{ref},i}\right)\right]. \tag{39}$$

Here $\ell_{\text{native},i}$ is the backbone's native MeanFlow or sCM loss, detailed below, and $\alpha_{\text{SFT}} = 10$. The expectation includes conditions, collected samples, and native-loss time and noise sampling. The selection mask weights only the native loss; the reference penalty applies to every collected sample. The collection model is periodically refreshed from the learner, while the base reference and velocity teacher remain frozen.

DiffusionNFT objective. Following DiffusionNFT (Zheng et al., 2026), the reward determines a weight between positive and negative regression. Using the same reward notation as the main text, we set $\mu_c = G^{-1}\sum_{k=1}^{G} R_k$, compute σ_{global} over the collected batch, and form $\widehat{A}_i = \text{clip}((R_i - \mu_c)/(\sigma_{\text{global}} + \epsilon_{\text{std}}), -A_{\max}, A_{\max})$. The NFT weight is $w_i = \frac{1}{2} + \widehat{A}_i/(2A_{\max})$, with $A_{\max} = 5$ and $\epsilon_{\text{std}} = 10^{-4}$. Let $F_{\theta_{\text{old}}}^{(i)}$ be the detached prediction of the collection model. The positive and negative predictions are

$$F_i^{\pm} = \text{sg}[F_{\theta_{\text{old}}}^{(i)}] \pm \beta_{\text{NFT}}\left(F_\theta^{(i)} - \text{sg}[F_{\theta_{\text{old}}}^{(i)}]\right), \qquad \widehat{x}_1^{(i,\pm)} = \mathcal{C}_i(F_i^{\pm}), \qquad \beta_{\text{NFT}} = 0.1. \tag{40}$$

Here $F_\theta^{(i)}$ denotes the raw backbone prediction, not the endpoint map f_t^θ in the main text. The reconstruction $\mathcal{C}_i$ makes this distinction explicit. For MeanFlow, use reverse-time coordinates $s = 1 - t$ and a sampled target time $s' \le s$. The noisy input is $x_s^{(i)} = (1-s)x_1^{(i)} + s\epsilon_i$, with $\epsilon_i \sim \mathcal{N}(0, I)$, and $F_\theta^{(i)} = \bar{u}_{s\to s'}^\theta(x_s^{(i)})$ is the reverse-time average velocity defined in Appendix C.2. NFT reconstructs $\mathcal{C}_i(F) = x_s^{(i)} - sF$. For sCM, let ω denote its native angular noise coordinate and set $x_\omega^{(i)} = \cos(\omega)x_1^{(i)} + \sigma_{\text{data}}\sin(\omega)\epsilon_i$. Its reconstruction is $\mathcal{C}_i(F) = \cos(\omega)x_\omega^{(i)} - \sigma_{\text{data}}\sin(\omega)F$, where $\sigma_{\text{data}} = 0.5$. The sCM network receives $x_\omega^{(i)}/\sigma_{\text{data}}$ and ω; $\mathcal{C}_i(F_\theta^{(i)})$ is its clean-endpoint prediction. Conditioning is suppressed in both expressions.

For each reconstruction, adaptive normalization uses the detached mean absolute error:

$$D(\widehat{x}, x) = \frac{d^{-1}\|\widehat{x} - x\|^2}{\max(\mathrm{sg}[d^{-1}\|\widehat{x} - x\|_1], 10^{-5})},$$
$$\mathcal{L}_{\mathrm{NFT}} = \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\left(w_i D(\widehat{x}_1^{(i,+)}, x_1^{(i)}) + (1 - w_i) D(\widehat{x}_1^{(i,-)}, x_1^{(i)}) + \alpha_{\mathrm{NFT}}\ell_{\mathrm{ref},i}\right)\right]. \tag{41}$$

where $\alpha_{\mathrm{NFT}} = 0.01$. Collected targets, reward weights, old-model predictions, and normalization factors are detached during each update. After each collection-and-update round, the old model is updated as $\theta_{\mathrm{old}} \leftarrow \eta_{\mathrm{old}}\theta_{\mathrm{old}} + (1 - \eta_{\mathrm{old}})\theta$, on the LoRA parameters, with the scheduled EMA decay η_{old}; the base reference remains frozen.

Native SFT retains an unadapted supervision field. Online SFT collects generated images, selects high-reward samples, and applies the backbone's native training loss together with a frozen-base reference penalty. This loss supervises more than the selected clean endpoint: it combines a velocity or tangent target with a JVP-based self-consistency condition. In our MeanFlow implementation, the instantaneous target comes from a frozen velocity teacher, while the JVP supplies the finite-interval correction. The sCM implementation likewise uses an independent frozen velocity teacher to define the tangent direction in its native consistency loss. Both use the pretrained rescaled FLUX velocity model; the frozen few-step base additionally provides reference regularization.

For the forward-time convention $t = 0$ for noise and $t = 1$ for data, the MeanFlow target combines the frozen instantaneous field u_t^{ϕ} with derivatives of the average-velocity prediction:

$$T_\theta(x, t, r) = u_t^{\phi}(x) + (r - t)\left[\partial_t u_{t\to r}^{\theta}(x) + D_x u_{t\to r}^{\theta}(x) u_t^{\phi}(x)\right],$$
$$\mathcal{L}_{\mathrm{MF}}(\theta) = \mathbb{E}\|u_{t\to r}^{\theta}(X_t) - \mathrm{sg}[T_\theta(X_t, t, r)]\|^2. \tag{42}$$

Here D_x is the spatial Jacobian and sg denotes stop-gradient. The target follows from the flow-map transport identity $\partial_t \psi_{t\to r} + D_x \psi_{t\to r} u_t = 0$. For the sCM endpoint predictor f_t^{θ}, the same identity yields the consistency residual $\partial_t f_t^{\theta} + D_x f_t^{\theta} u_t^{\phi}$, with $f_1(x) = x$ (Lu & Song, 2025). Its squared residual, with the native weighting and stabilization, couples endpoint learning to the teacher's tangent direction. These relations identify the dependence on the unadapted instantaneous field in both native losses.

Reward selection changes the endpoint distribution but does not update this teacher. To see the resulting mismatch, let $p_{\mathrm{sel}}(x_1 \mid c)$ denote the reward-selected distribution and use the linear path $x_t = tx_1 + (1 - t)\epsilon$, with $\epsilon \sim \mathcal{N}(0, I)$. The instantaneous field associated with these endpoints is

$$u_{\mathrm{sel}}(x_t, t, c) = \mathbb{E}_{x_1 \sim p_{\mathrm{sel}}}[x_1 - \epsilon \mid x_t, t, c]. \tag{43}$$

The frozen teacher instead approximates a field associated with its original pretraining distribution. In general, reward selection changes the conditional expectation above, so the two fields need not agree. Increasing the frequency of high-reward endpoints therefore does not supply a correspondingly adapted instantaneous target. The JVP/consistency term continues to couple the student to the frozen teacher's dynamics, and the base-reference penalty further discourages departure from the pretrained predictor. Consequently, the unchanged native objective is not, in general, a consistent distillation objective for the reward-selected distribution: the endpoint samples and supervisory dynamics refer to different distributions. Reward filtering changes the endpoint distribution without updating the teacher's instantaneous field, leaving the native flow-map loss with inconsistent supervision.

Flow-map outputs differ from forward-noising targets. The distributional interpretation of DiffusionNFT relies on instantaneous-velocity identities (Theorems 3.1–3.2 and Appendix A of Zheng et al., 2026). Let $p_{\mathrm{old}}(x_1 \mid c)$ be the collected endpoint distribution and $w(x_1, c) \in [0, 1]$ its reward-derived weight. The positive distribution is $p_+(x_1 \mid c) = w(x_1, c)p_{\mathrm{old}}(x_1 \mid c)/Z_c$, where $Z_c = \mathbb{E}_{p_{\mathrm{old}}}[w(X_1, c)]$; the negative distribution is defined analogously using $1 - w$. Their proof converts a mixture identity for the corresponding posteriors into an identity for their instantaneous velocity fields, using the affine relation between instantaneous velocity and posterior mean. This establishes NFT's reinforcement-guidance direction; an appropriate guidance strength recovers the positive velocity field and hence p_+. Thus, applying this argument requires the predictor to

represent the instantaneous velocity, or an equivalent denoising quantity satisfying the same identity. Substituting a long-range average velocity or transported endpoint does not meet that requirement and therefore does not inherit the claimed connection between the regression objective and the reward-preferred distribution.

DiffusionNFT (Zheng et al., 2026) builds on forward-noising regression, whose prediction target is determined by the conditional distribution at a single noise level. To make the distinction explicit, consider the underlying squared-error regression with $x_s = (1-s)x_1 + s\epsilon$ in reverse-time coordinates $s = 1 - t$. Its instantaneous-velocity target is $\epsilon - x_1$, so the population minimizer is $u_s^*(x_s) = \mathbb{E}[\epsilon - x_1 \mid x_s, c]$. Consequently, $x_s - s u_s^*(x_s) = \mathbb{E}[x_1 \mid x_s, c]$: the corresponding reconstruction is a posterior-mean denoiser. These identities describe the prediction quantities underlying the regression, rather than asserting an unchanged minimizer for the reward-weighted NFT objective.

MeanFlow instead learns the interval-average velocity $\bar{u}^\theta_{s\to s'}(x_s)$ defining the long-range transport $\psi^\theta_{s\to s'}(x_s) = x_s - (s - s')\bar{u}^\theta_{s\to s'}(x_s)$. Only the equal-time limit $s' \to s$ recovers the instantaneous velocity; a finite-interval average generally differs from $u_s(x_s)$. In the NFT reconstruction in Equation (40), using this average as $x_s - s\bar{u}^\theta_{s\to s'}(x_s)$ therefore subjects a long-range prediction to an instantaneous-prediction objective. The distinction is not merely the coefficient s versus $s - s'$: even when $s' = 0$, the transported endpoint $\psi_{s\to 0}(x_s)$ need not equal $\mathbb{E}[x_1 \mid x_s, c]$. Transport follows the marginal dynamics, whereas the conditional mean averages the clean samples compatible with a noisy observation. Replacing the coefficient alone cannot reconcile these quantities; the flow-map transport identity introduces the derivative correction shown in Equation (42).

The same distinction applies to sCM. Its endpoint prediction $f_t^\theta(x_t)$ approximates the endpoint of a trajectory, while forward-noising regression targets a conditional denoising statistic. For an accurate flow map, $f_t(X_t)$ has the data marginal when $X_t \sim p_t$; a posterior mean generally has reduced variability because it averages over uncertain clean endpoints. For example, under the law of total covariance, $\mathrm{Cov}(X_1) = \mathrm{Cov}(\mathbb{E}[X_1 \mid X_t]) + \mathbb{E}[\mathrm{Cov}(X_1 \mid X_t)]$. With nonzero posterior uncertainty and finite second moments, the posterior mean therefore cannot have the same covariance as the data. Thus, a flow-map endpoint and a posterior-mean denoiser are generally different functions, even when both are written as clean-sample predictions. NFT's positive/negative regression and adaptive normalization in Equation (41) do not establish the missing identity between them. This mismatch arises from the quantities represented by flow maps and requires a compatible objective, rather than a change to shared optimization settings.

Relation to RWR and DPO. Reward-weighted regression (RWR) (Lee et al., 2023) and Diffusion-DPO (Wallace et al., 2023) also use local denoising losses, respectively as a weighted fitting objective and a diffusion-likelihood surrogate. Average-velocity and consistency-map outputs do not generally satisfy the conditional-prediction identities underlying these losses. This observation concerns their standard diffusion implementations, rather than RWR or DPO as general optimization principles. We select DiffusionNFT as a representative of this family based on its reported performance in diffusion post-training (Zheng et al., 2026).

Direct optimization of the flow-map policy. Flow-Map GRPO resolves both dependencies by defining a stochastic policy around the pretrained flow map and optimizing its tractable transition likelihood ratios using rollout rewards. The update acts on the long-range generator used by the policy, without requiring a separately reward-adapted instantaneous teacher or identifying the flow-map output with a local denoising statistic. This is the relevant distinction behind the comparison in Table 3.

C.5 OCR and PickScore Training Curves

Figure 7 compares SFT, NFT, and Flow-Map GRPO using full-test rewards evaluated every 20 training steps with EMA weights and five sampler steps ($K = 4$). OCR runs are compared through step 700 for sCM and 800 for MeanFlow; PickScore runs extend to step 1,500. The curves show training-time evaluations without additional smoothing. Dataset sizes and evaluation conventions are given in Appendix C.2.

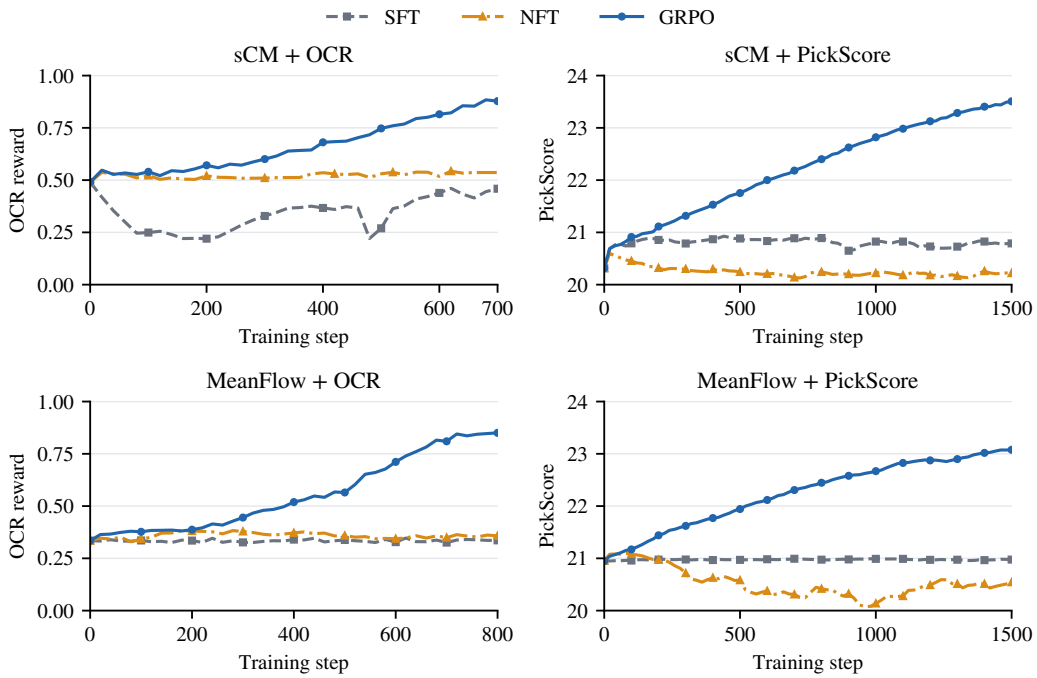


Figure 7: Training-time EMA full-evaluation rewards for sCM (top) and MeanFlow (bottom), optimizing OCR (left) or PickScore (right). OCR is shown through steps 700 and 800, respectively; both PickScore axes extend to 1500. GRPO curves share the corresponding SFT initialization reference.

D Complete Text-to-Image Results

We evaluate the post-trained models across multiple numbers of sampling steps to assess transfer to inference settings that differ from the five-step training sampler ($K = 4$). The same EMA adapters are evaluated without further training and compared with Base at each inference step count. Task Metrics measure RL improvements on the target tasks using their respective test sets. DrawBench Metrics are auxiliary general-quality metrics used to monitor degradation outside the optimized reward. UniRwd and DeQA use raw 1–5 scores for both backbones. DB Sum denotes the unweighted sum of the five displayed quality metrics. All numbers are single-run point estimates.

D.1 OCR-Only Evaluations

Table 9: Evaluation of post-trained MeanFlow at different numbers of sampling steps (OCR-only reward). Task Metrics measure RL improvements on the target tasks; DrawBench Metrics are auxiliary general-quality metrics used to monitor degradation outside the optimized reward. Evaluation settings are given in Appendix C.2.

K	Model	Task Metrics			DrawBench Metrics					
		GenEval↑	OCR Acc.↑	PickScore Task↑	DB-PickScore↑	Aesthetic↑	DeQA↑	ImgRwd↑	UniRwd↑	DB Sum↑
3	MeanFlow Base	–	0.2876	–	21.3577	**5.4391**	**4.0001**	0.2157	2.5495	33.5621
3	MeanFlow + Flow-Map GRPO LoRA	–	**0.7730**	–	**21.5779**	5.3550	3.8437	**0.3076**	**2.5688**	**33.6530**
4	MeanFlow Base	–	0.3491	–	21.6298	**5.5368**	**4.1712**	0.3918	2.7187	34.4483
4	MeanFlow + Flow-Map GRPO LoRA	–	**0.8504**	–	**21.9869**	5.5111	4.1396	**0.5386**	**2.8049**	**34.9812**
7	MeanFlow Base	–	0.4517	–	22.0761	5.6090	4.2500	0.5744	2.9216	35.4311
7	MeanFlow + Flow-Map GRPO LoRA	–	**0.9256**	–	**22.4061**	**5.6454**	**4.3415**	**0.7855**	**3.0262**	**36.2047**
15	MeanFlow Base	–	0.5507	–	22.2302	5.6489	4.3217	0.6870	3.0208	35.9086
15	MeanFlow + Flow-Map GRPO LoRA	–	**0.9385**	–	**22.5182**	**5.6824**	**4.4106**	**0.8755**	**3.0968**	**36.5835**

Table 10: Evaluation of post-trained sCM at different numbers of sampling steps (OCR-only reward). Task Metrics measure RL improvements on the target tasks; DrawBench Metrics are auxiliary general-quality metrics used to monitor degradation outside the optimized reward. Evaluation settings are given in Appendix C.2.

K	Model	Task Metrics			DrawBench Metrics					
		GenEval↑	OCR Acc.↑	PickScore Task↑	DB-PickScore↑	Aesthetic↑	DeQA↑	ImgRwd↑	UniRwd↑	DB Sum↑
2	sCM Base	–	0.3394	–	**21.4149**	**5.3850**	**3.4327**	**0.5623**	**2.6225**	**33.4174**
	sCM + Flow-Map GRPO LoRA	–	**0.7608**	–	21.1626	5.1030	3.1430	0.1033	2.4512	31.9631
3	sCM Base	–	0.4290	–	21.4524	**5.4448**	**3.6554**	**0.5742**	**2.6727**	**33.7994**
	sCM + Flow-Map GRPO LoRA	–	**0.8592**	–	**21.4766**	5.3037	3.5807	0.2819	2.6603	33.3033
4	sCM Base	–	0.4903	–	21.4993	**5.4747**	3.7469	**0.5750**	2.6725	**33.9684**
	sCM + Flow-Map GRPO LoRA	–	**0.8863**	–	**21.5647**	5.4049	**3.7800**	0.3475	**2.7137**	33.8107
7	sCM Base	–	0.6120	–	21.5128	5.5084	3.8796	**0.6037**	2.7196	34.2241
	sCM + Flow-Map GRPO LoRA	–	**0.8825**	–	**21.6504**	**5.5197**	**4.0625**	0.4857	**2.8560**	**34.5743**
15	sCM Base	–	0.6583	–	21.3638	5.5358	3.8862	**0.5312**	2.6591	33.9761
	sCM + Flow-Map GRPO LoRA	–	**0.8988**	–	**21.5153**	**5.5614**	**4.1601**	0.4618	**2.7936**	**34.4922**

D.2 PICKSCORE-ONLY EVALUATIONS

These two tables report the MeanFlow and sCM PickScore-only adapters with EMA weights at training step 1,500. Base rows use the evaluation setting in Appendix C.2. Both backbones use raw 1–5 UniRwd scores, and DB Sum denotes the unweighted sum of the five DrawBench metrics. Each value K corresponds to $K + 1$ sampling steps.

Table 11: Evaluation of post-trained MeanFlow at different numbers of sampling steps (PickScore-only reward). Task Metrics measure RL improvements on the target tasks; DrawBench Metrics are auxiliary general-quality metrics used to monitor degradation outside the optimized reward. Evaluation settings are given in Appendix C.2.

K	Model	Task Metrics			DrawBench Metrics					
		GenEval↑	OCR Acc.↑	PickScore Task↑	DB-PickScore↑	Aesthetic↑	DeQA↑	ImgRwd↑	UniRwd↑	DB Sum↑
3	MeanFlow Base	–	–	20.6926	21.3577	5.4391	**4.0001**	0.2157	2.5495	33.5621
	MeanFlow + Flow-Map GRPO LoRA	–	–	**22.7035**	**22.8560**	**5.9958**	3.7091	**1.0043**	**3.0077**	**36.5730**
4	MeanFlow Base	–	–	20.9758	21.6298	5.5368	**4.1712**	0.3918	2.7187	34.4483
	MeanFlow + Flow-Map GRPO LoRA	–	–	**23.0772**	**23.3711**	**6.1510**	4.0829	**1.1402**	**3.2005**	**37.9456**
7	MeanFlow Base	–	–	21.3595	22.0761	5.6090	4.2500	0.5744	2.9216	35.4311
	MeanFlow + Flow-Map GRPO LoRA	–	–	**23.2963**	**23.6881**	**6.2571**	**4.3427**	**1.1950**	**3.3168**	**38.7997**
15	MeanFlow Base	–	–	21.4382	22.2302	5.6489	4.3217	0.6870	3.0208	35.9086
	MeanFlow + Flow-Map GRPO LoRA	–	–	**23.3229**	**23.7649**	**6.2959**	**4.4283**	**1.2246**	**3.3445**	**39.0582**

Table 12: Evaluation of post-trained sCM at different numbers of sampling steps (PickScore-only reward). Task Metrics measure RL improvements on the target tasks; DrawBench Metrics are auxiliary general-quality metrics used to monitor degradation outside the optimized reward. Evaluation settings are given in Appendix C.2.

K	Model	Task Metrics			DrawBench Metrics					
		GenEval↑	OCR Acc.↑	PickScore Task↑	DB-PickScore↑	Aesthetic↑	DeQA↑	ImgRwd↑	UniRwd↑	DB Sum↑
2	sCM Base	–	–	20.3023	21.4149	5.3850	3.4327	0.5623	2.6225	33.4174
	sCM + Flow-Map GRPO LoRA	–	–	**23.2390**	**23.5695**	**6.0445**	**3.7773**	**1.1965**	**3.2104**	**37.7981**
3	sCM Base	–	–	20.2942	21.4524	5.4448	3.6554	0.5742	2.6727	33.7994
	sCM + Flow-Map GRPO LoRA	–	–	**23.4376**	**23.7651**	**6.0433**	**4.0933**	**1.2456**	**3.3186**	**38.4659**
4	sCM Base	–	–	20.3517	21.4993	5.4747	3.7469	0.5750	2.6725	33.9684
	sCM + Flow-Map GRPO LoRA	–	–	**23.4820**	**23.8084**	**6.0360**	**4.2343**	**1.2600**	**3.3361**	**38.6748**
7	sCM Base	–	–	20.2940	21.5128	5.5084	3.8796	0.6037	2.7196	34.2241
	sCM + Flow-Map GRPO LoRA	–	–	**23.4160**	**23.7285**	**6.0048**	**4.3955**	**1.2236**	**3.3273**	**38.6797**
15	sCM Base	–	–	20.0556	21.3638	5.5358	3.8862	0.5312	2.6591	33.9761
	sCM + Flow-Map GRPO LoRA	–	–	**23.0871**	**23.4470**	**5.9367**	**4.4677**	**1.1595**	**3.2983**	**38.3090**

D.3 GENEVAL EVALUATIONS

Table 13: Evaluation of post-trained MeanFlow at different numbers of sampling steps (GenEval reward). Task Metrics measure RL improvements on the target tasks; DrawBench Metrics are auxiliary general-quality metrics used to monitor degradation outside the optimized reward. Evaluation settings are given in Appendix C.2.

K	Model	Task Metrics			DrawBench Metrics					
		GenEval↑	OCR Acc.↑	PickScore Task↑	DB-PickScore↑	Aesthetic↑	DeQA↑	ImgRwd↑	UniRwd↑	DB Sum↑
3	MeanFlow Base	0.4684	–	–	21.3577	**5.4391**	**4.0001**	0.2157	2.5495	**33.5621**
3	MeanFlow + Flow-Map GRPO LoRA	**0.7938**	–	–	**21.4279**	5.2559	2.8478	**0.3979**	**2.6137**	32.5431
4	MeanFlow Base	0.5041	–	–	21.6298	**5.5368**	**4.1712**	0.3918	2.7187	**34.4483**
4	MeanFlow + Flow-Map GRPO LoRA	**0.8454**	–	–	**21.8184**	5.3905	3.2086	**0.6426**	**2.8719**	33.9319
7	MeanFlow Base	0.5358	–	–	22.0761	**5.6090**	**4.2500**	0.5744	2.9216	**35.4311**
7	MeanFlow + Flow-Map GRPO LoRA	**0.8683**	–	–	**22.2332**	5.5216	3.6592	**0.8449**	**3.0929**	35.3517
15	MeanFlow Base	0.5518	–	–	22.2302	**5.6489**	**4.3217**	0.6870	3.0208	**35.9086**
15	MeanFlow + Flow-Map GRPO LoRA	**0.8789**	–	–	**22.3262**	5.5534	3.9166	**0.8904**	**3.1842**	35.8708

Table 14: Evaluation of post-trained sCM at different numbers of sampling steps (GenEval reward). Task Metrics measure RL improvements on the target tasks; DrawBench Metrics are auxiliary general-quality metrics used to monitor degradation outside the optimized reward. Evaluation settings are given in Appendix C.2.

K	Model	Task Metrics			DrawBench Metrics					
		GenEval↑	OCR Acc.↑	PickScore Task↑	DB-PickScore↑	Aesthetic↑	DeQA↑	ImgRwd↑	UniRwd↑	DB Sum↑
2	sCM Base	0.4797	–	–	21.4149	**5.3850**	**3.4327**	0.5623	2.6225	33.4174
2	sCM + Flow-Map GRPO LoRA	**0.7721**	–	–	**21.9073**	5.0662	3.2860	**0.6124**	**2.9040**	**33.7758**
3	sCM Base	0.4560	–	–	21.4524	**5.4448**	3.6554	0.5742	2.6727	33.7994
3	sCM + Flow-Map GRPO LoRA	**0.8005**	–	–	**22.1208**	5.1971	**3.6765**	**0.6813**	**3.0374**	**34.7131**
4	sCM Base	0.4513	–	–	21.4993	**5.4747**	3.7469	0.5750	2.6725	33.9684
4	sCM + Flow-Map GRPO LoRA	**0.8049**	–	–	**22.1046**	5.2401	**3.8938**	**0.6514**	**3.0835**	**34.9734**
7	sCM Base	0.4299	–	–	21.5128	**5.5084**	3.8796	0.6037	2.7196	34.2241
7	sCM + Flow-Map GRPO LoRA	**0.7968**	–	–	**22.1326**	5.3483	**4.1705**	**0.6881**	**3.1347**	**35.4741**
15	sCM Base	0.3983	–	–	21.3638	**5.5358**	3.8862	0.5312	2.6591	33.9761
15	sCM + Flow-Map GRPO LoRA	**0.7585**	–	–	**21.8789**	5.3724	**4.2218**	**0.5578**	**3.0663**	**35.0972**

D.4 JOINT OCR–PICKSCORE EVALUATIONS

Table 15: Evaluation of post-trained MeanFlow at different numbers of sampling steps (joint OCR–PickScore rewards). Task Metrics measure RL improvements on the target tasks; DrawBench Metrics are auxiliary general-quality metrics used to monitor degradation outside the optimized reward. Evaluation settings are given in Appendix C.2.

K	Model	Task Metrics			DrawBench Metrics					
		GenEval↑	OCR Acc.↑	PickScore Task↑	DB-PickScore↑	Aesthetic↑	DeQA↑	ImgRwd↑	UniRwd↑	DB Sum↑
3	MeanFlow Base	–	0.2876	20.6926	21.3577	5.4391	**4.0001**	0.2157	2.5495	33.5621
3	MeanFlow + Flow-Map GRPO LoRA	–	**0.6275**	**22.1122**	**22.5958**	**5.8244**	3.8551	**0.8696**	**3.0020**	**36.1468**
4	MeanFlow Base	–	0.3491	20.9758	21.6298	5.5368	4.1712	0.3918	2.7187	34.4483
4	MeanFlow + Flow-Map GRPO LoRA	–	**0.7719**	**22.4964**	**23.0506**	**5.9299**	**4.1892**	**1.0439**	**3.1876**	**37.4012**
7	MeanFlow Base	–	0.4517	21.3595	22.0761	5.6090	4.2500	0.5744	2.9216	35.4311
7	MeanFlow + Flow-Map GRPO LoRA	–	**0.8701**	**22.7455**	**23.4005**	**6.0176**	**4.3873**	**1.1342**	**3.3354**	**38.2750**
15	MeanFlow Base	–	0.5507	21.4382	22.2302	5.6489	4.3217	0.6870	3.0208	35.9086
15	MeanFlow + Flow-Map GRPO LoRA	–	**0.8845**	**22.7747**	**23.4243**	**6.0410**	**4.4403**	**1.1352**	**3.3692**	**38.4101**

Table 16: Evaluation of post-trained sCM at different numbers of sampling steps (joint OCR–PickScore rewards). Task Metrics measure RL improvements on the target tasks; DrawBench Metrics are auxiliary general-quality metrics used to monitor degradation outside the optimized reward. Evaluation settings are given in Appendix C.2.

K **Model**	**Task Metrics**			**DrawBench Metrics**					
	GenEval↑	**OCR Acc.↑**	**PickScore Task↑**	**DB-PickScore↑**	**Aesthetic↑**	**DeQA↑**	**ImgRwd↑**	**UniRwd↑**	**DB Sum↑**
2 sCM Base	–	0.3394	20.3023	21.4149	5.3850	3.4327	0.5623	2.6225	33.4174
2 sCM + Flow-Map GRPO LoRA	–	**0.7744**	**22.3077**	**22.8201**	**5.8086**	**3.7308**	**0.9311**	**2.9627**	**36.2533**
3 sCM Base	–	0.4290	20.2942	21.4524	5.4448	3.6554	0.5742	2.6727	33.7994
3 sCM + Flow-Map GRPO LoRA	–	**0.8380**	**22.4789**	**23.0224**	**5.9183**	**4.1679**	**0.9719**	**3.0648**	**37.1452**
4 sCM Base	–	0.4903	20.3517	21.4993	5.4747	3.7469	0.5750	2.6725	33.9684
4 sCM + Flow-Map GRPO LoRA	–	**0.8568**	**22.5153**	**23.0403**	**5.9494**	**4.3311**	**0.9741**	**3.1004**	**37.3952**
7 sCM Base	–	0.6120	20.2940	21.5128	5.5084	3.8796	0.6037	2.7196	34.2241
7 sCM + Flow-Map GRPO LoRA	–	**0.8592**	**22.3719**	**22.9681**	**5.9591**	**4.4511**	**0.9381**	**3.0669**	**37.3833**
15 sCM Base	–	0.6583	20.0556	21.3638	5.5358	3.8862	0.5312	2.6591	33.9761
15 sCM + Flow-Map GRPO LoRA	–	**0.8713**	**21.9501**	**22.6570**	**5.9168**	**4.4899**	**0.8496**	**2.9970**	**36.9102**

E Additional Qualitative Results

These examples use the adapters and numbers of inference steps in the quantitative tables. Each prompt and seed is matched across Base (top) and Ours (bottom) and across the step sweep. Columns show K, corresponding to $K + 1$ sampler steps.

E.1 MEANFLOW: OCR

Prompt 1: A premium editorial photograph of a frosted peach soda can splashing into turquoise water, sliced peaches suspended nearby. The can label clearly reads “PEACH CLUB”. Crisp droplets, direct summer sunlight, clean commercial composition.



Prompt 2: A sophisticated illustrated book cover with golden koi fish swimming through dark indigo waves and floating white petals. Large elegant ivory title “THE SILENT SEA”. Intricate woodblock illustration, refined spacing.



Prompt 3: A sophisticated luxury product photograph of a transparent amber glass bottle with label “GOLDEN HOUR”, placed on a moss-covered rock beside a mountain stream. Sunlight through the glass, emerald water, delicate vapor.



Figure 8: MeanFlow before and after OCR post-training. Three prompts, each shown as a paired Base/Ours sweep at different numbers of sampling steps.

E.2 sCM: OCR

Prompt 1: A cinematic photograph of a cherry-red racing car beside a desert gas station at blue hour. A tall illuminated sign clearly reads “LAST STOP”. Pink clouds, cyan shadows, chrome highlights, expansive composition.



Prompt 2: A beautiful interior photograph of a coral-pink greenhouse filled with white orchids and ferns. A prominent cream wall sign reads “GROW WITH LOVE”. Warm sunlight, patterned shadows, polished green tile floor.



Prompt 3: A close-up commercial photograph of a jade-green matcha bottle standing among bright green leaves on pale pink stone. Large white label reads “MATCHA MORNING”. Dewdrops, soft luminous backlighting, clean sophisticated composition.



Figure 9: sCM before and after OCR post-training. Three prompts, each shown as a paired Base/Ours sweep at different numbers of sampling steps.

E.3 MEANFLOW: PICKSCORE

Prompt 1: A cinematic macro photograph of a tiny red frog sitting inside a translucent white orchid, rain droplets, luminous tropical foliage, very shallow depth of field.



Prompt 2: A whimsical miniature village carved inside a giant orange pumpkin, tiny glowing windows and wooden bridges, autumn leaves, richly detailed stop-motion film aesthetic.



Prompt 3: A transparent crystal whale floating above a mirror-flat ocean at twilight, stars refracting through its body, a tiny boat below, breathtaking cinematic fantasy realism.



Figure 10: MeanFlow before and after PickScore post-training. Three prompts, each shown as a paired Base/Ours sweep at different numbers of sampling steps.

E.4 sCM: PickScore

Prompt 1: A dramatic editorial photograph of a scarlet sailing yacht surrounded by enormous turquoise icebergs, dark storm clouds opening to a shaft of golden sunlight, fine windblown spray, sweeping aerial composition.



Prompt 2: A beautifully composed food photograph of a raspberry pavlova with cream and rose petals on a mint-green ceramic plate, soft side light, fine sugar dust, luxurious editorial styling.



Prompt 3: A dramatic photograph of a massive curling ocean wave seen from inside the barrel, translucent aquamarine water and a golden setting sun, crisp foam and mist.

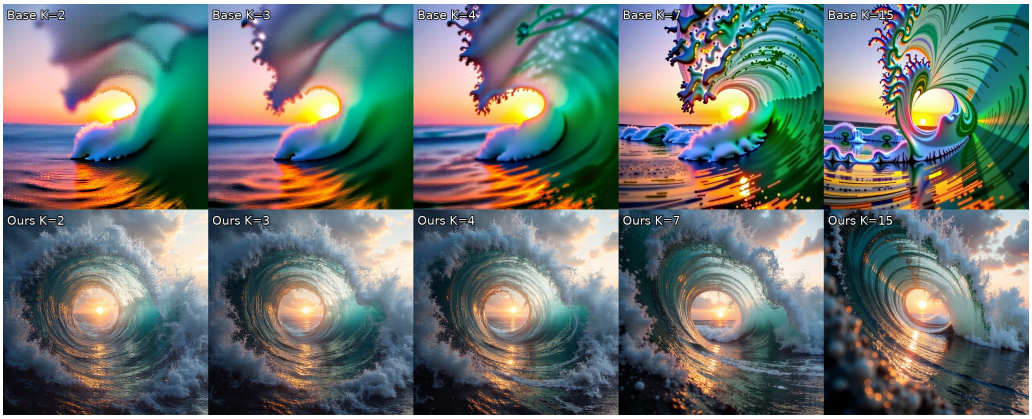


Figure 11: sCM before and after PickScore post-training. Three prompts, each shown as a paired Base/Ours sweep at different numbers of sampling steps.

E.5 MEANFLOW: GENEVAL

1. a photo of a backpack left of a chair



2. a photo of a teddy bear right of a bowl



3. a photo of a bottle left of a clock



4. a photo of three teddy bears



5. a photo of a red chair and a blue teddy bear



6. a photo of three vases



7. a photo of a blue bowl and a red cup



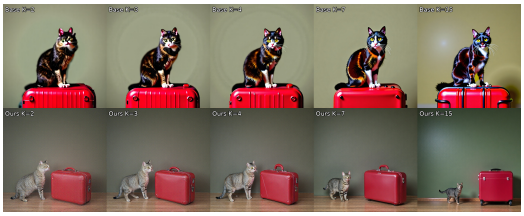
8. a photo of a pineapple and a camera



Figure 12: MeanFlow before and after GenEval post-training. Eight prompt groups illustrate object presence, counting, color, and spatial relations.

E.6 sCM: GenEval

1. a photo of a cat left of a red suitcase



2. a photo of a vase left of a laptop



3. a photo of a teddy bear right of a bowl



4. a photo of three teddy bears



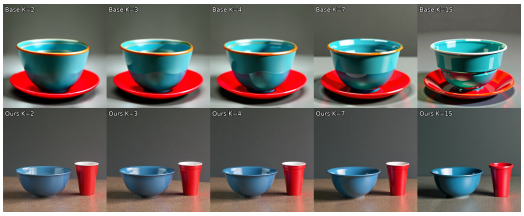
5. a photo of three vases



6. a photo of four cups



7. a photo of a blue bowl and a red cup



8. a photo of a violin and a suitcase



Figure 13: sCM before and after GenEval post-training. Eight prompt groups illustrate object presence, counting, color, and spatial relations.

E.7 MeanFlow: OCR + PickScore

Prompt 1: A fantastical cinematic photograph of a floating pastry shop suspended beneath a cream balloon above lavender fields. Its large storefront sign reads “SUGAR CLOUD”. Golden sunrise, tiny intricate pastries, beautiful pastel color palette.



Prompt 2: A premium dark chocolate wrapper arranged with raspberries and broken chocolate on rose marble. Large elegant gold lettering reads “DARK BLOOM”. Burgundy paper, hard editorial shadows, exquisite food photography.



Prompt 3: A beautiful illustrated cover of a cookbook, lemons, sardines and ceramic bowls arranged around a large cream central panel. The title reads “A TASTE OF SUMMER”. Mediterranean blue and yellow palette, textured gouache painting.



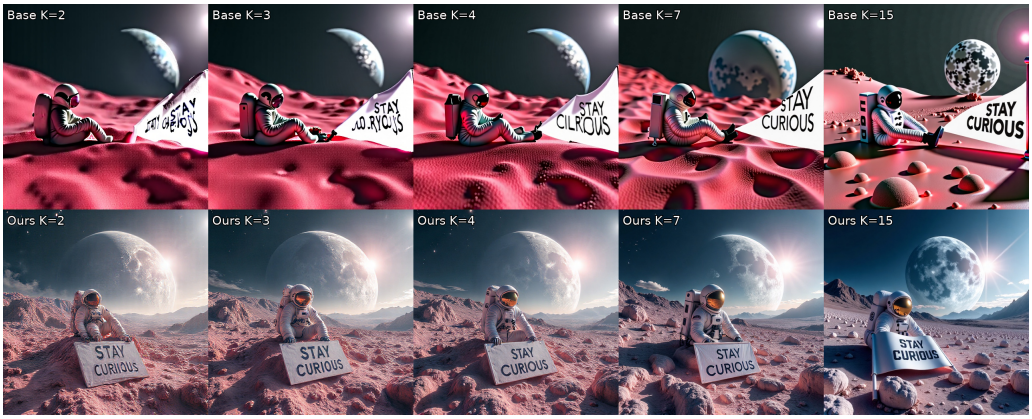
Figure 14: MeanFlow before and after OCR + PickScore post-training. Three prompts, each shown as a paired Base/Ours sweep at different numbers of sampling steps.

E.8 sCM: OCR + PickScore

Prompt 1: A striking editorial photograph of a woman in a flowing red dress standing beside a giant white hot-air balloon on a salt flat at sunrise. The balloon clearly reads “DREAM BIG” in large black letters. Pink sky reflected in shallow water, strong graphic composition.



Prompt 2: A cinematic wide view of an astronaut resting on a pink lunar ridge, Earth in the dark sky. A white flag in the foreground reads “STAY CURIOUS” in clear dark lettering. Detailed spacesuit, dramatic rim lighting, spectacular landscape.



Prompt 3: A moody editorial photograph of a burgundy vinyl record sleeve beside a vintage turntable, a single red rose and warm amber light. Large cream title on the sleeve reads “MIDNIGHT JAZZ”. Rich tactile paper and wood textures.



Figure 15: sCM before and after OCR + PickScore post-training. Three prompts, each shown as a paired Base/Ours sweep at different numbers of sampling steps.