MRFT: MEANFLOW REINFORCEMENT FINE-TUNING VIA TRACTABLE STOCHASTIC TRANSITIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Flow matching has shown strong performance in continuous robotic control, but its multi-step sampling procedures incur substantial rollout costs in online reinforcement learning (RL). MeanFlow presents a compelling solution to this sampling overhead by enabling efficient few-step generation through explicitly modeling the average velocity field. However, directly applying MeanFlow to online RL is hindered by a critical limitation: its deterministic sampling process lacks the stochastic exploration and tractable transition likelihoods required for RL fine-tuning. Existing techniques depend on auxiliary learnable networks for post-hoc stochasticity injection, inherently increasing computational overhead and parameter complexity. Rather than relying on heuristic designs, we formally derive **MeanFlow-SDE** from the underlying flow dynamics. By approximating the finite-interval integration of the marginal-preserving flow SDE, MeanFlow-SDE yields tractable Gaussian transitions that inherently support stochastic exploration, eliminating the need for additional components. Building upon MeanFlow-SDE, we introduce **MeanFlow Reinforcement Fine-Tuning (MRFT)**, an efficient online fine-tuning framework tailored for MeanFlow policies. By embedding our tractable transitions into RL objectives, MRFT seamlessly integrates stochastic generative sampling with robust policy optimization. Extensive evaluations on lightweight control policies, $\pi_{0.5}$-style architectures, and real-world robotic systems demonstrate consistent improvements in sample efficiency and task success rates while maintaining few-step inference.

1 INTRODUCTION

In recent years, flow matching (Lipman et al., 2023) has demonstrated strong performance in continuous robotic control, benefiting from stable training and the capability to model multi-modal action distributions (Chisari et al., 2024). However, its multi-step sampling procedures incur significant latency and computational overhead per action (Sochopoulos et al., 2025; Chi et al., 2023). This computational burden becomes particularly pronounced in online reinforcement learning (RL), where action generation is repeatedly executed during the rollout phase (Ren et al., 2025; Park et al., 2025). These limitations make few-step generative models appealing for online robotic RL, as they reduce the number of sampling steps while maintaining high-quality action generation.

While several few-step flow-based policies have been proposed—such as AdaFlow (Hu et al., 2024), FlowPolicy (Zhang et al., 2025a) and ManiFlow (Yan et al., 2025)—they often rely on complex adaptive integrations or auxiliary consistency-style training objectives (Prasad et al., 2024). In contrast, MeanFlow offers a more elegant alternative by directly estimating the average velocity over a finite interval, bypassing the need for instantaneous velocity prediction at every timestep. This formulation enables high-quality, few-step generation without the complexities of distillation or curriculum training (Geng et al., 2025; Sheng et al., 2026), rendering MeanFlow exceptionally well-suited for low-latency continuous control.

However, directly applying MeanFlow to online RL is non-trivial. Online RL typically relies on stochastic sampling to explore the environment, whereas MeanFlow defines a deterministic finite-interval sampling process (Ren et al., 2025; Geng et al., 2025). Existing techniques address this via auxiliary learnable networks for post-hoc stochasticity injection (Zou et al., 2026). While functionally viable, this approach inherently inflates parameter complexity and introduces additional compu-

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.

tational overhead during fine-tuning. This raises a natural question: *Can we construct a simple and principled stochastic MeanFlow transition for online RL without introducing additional learnable components?*

To answer this question, we start from the observation that MeanFlow does not alter the underlying flow dynamics; instead, it represents deterministic transport over a finite interval through the average velocity defined along the corresponding flow trajectory. This motivates introducing stochasticity through the marginal-preserving flow SDE (Liu et al., 2025). To obtain a stochastic transition compatible with MeanFlow sampling, we integrate this SDE over a finite interval. The resulting exact representation still involves integrals over the intermediate stochastic trajectory and thus cannot be evaluated directly from the interval endpoints. We reduce these path-dependent terms analytically to isolate a finite-interval velocity integral, which can then be approximated using the MeanFlow average-velocity definition. This yields **MeanFlow-SDE**, a tractable Gaussian finite-interval transition that retains few-step sampling while approximating the marginal-preserving flow SDE with a bounded residual error.

Building on MeanFlow-SDE, we introduce **MeanFlow Reinforcement Fine-Tuning (MRFT)**, a framework for online reinforcement fine-tuning of few-step MeanFlow policies. MRFT treats the sequence of Gaussian transitions as a stochastic sampling process, providing tractable likelihoods over the resulting sampling paths for policy optimization. This framework enables effective exploration and reinforcement fine-tuning without introducing auxiliary learnable components, while preserving the few-step generation efficiency of MeanFlow.

Our main contributions are summarized as follows:

- We derive **MeanFlow-SDE**, a principled stochastic sampling formulation of MeanFlow. By analytically reducing the exact finite-interval integral of the marginal-preserving flow SDE to isolate its velocity integral, we enable direct MeanFlow approximation, yielding tractable Gaussian transitions without auxiliary learnable components.
- We propose **MeanFlow Reinforcement Fine-Tuning (MRFT)**, which uses Gaussian MeanFlow-SDE transitions to construct tractable sampling-path likelihoods for online RL, enabling efficient exploration and both on-policy and off-policy optimization while retaining few-step action generation.
- We evaluate MRFT in diverse simulated and real-world settings, scaling from lightweight continuous control to a $\pi_{0.5}$-style vision-language-action architecture. MRFT consistently outperforms pre-trained policies, boosting manipulation success rates in Franka Kitchen and Robomimic by an average of 72.8%, while simultaneously improving locomotion rewards by 57.1%.

2 Related Work

2.1 Few-Step Generative Policies for Robotic Control

Diffusion Policy (Chi et al., 2023) pioneered visuomotor control via denoising diffusion, with DP3 (Ze et al., 2024) extending to 3D inputs. Flow Matching and Rectified Flow provide an alternative by learning continuous velocity fields (Lipman et al., 2023; Liu et al., 2023), applied in AdaFlow (Hu et al., 2024) and FlowPolicy (Zhang et al., 2025a). To reduce the iterative sampling cost of these generative policies, recent work has explored diffusion distillation (Prasad et al., 2024; Wang et al., 2025) and direct one-step or few-step flow training (Frans et al., 2025; Yan et al., 2025). These approaches substantially reduce the number of function evaluations (NFE) required for action generation, motivating few-step generative policies for efficient robotic control.

MeanFlow provides a particularly direct formulation for this goal by learning the average velocity over a finite time interval, enabling distillation-free few-step generation (Geng et al., 2025). Building on this formulation, MP1 first applies MeanFlow to 3D robotic manipulation (Sheng et al., 2026). Subsequent methods improve MeanFlow policies through dispersive representation regularization (Zou et al., 2025), directional alignment (Fang et al., 2025), and instantaneous-velocity constraints (Zhan et al., 2026), further strengthening few-step action generation. Collectively, these studies establish MeanFlow as an emerging low-NFE backbone for robotic policies.

2.2 Online Reinforcement Learning with Flow-Based Policies

Online reinforcement learning has increasingly been used to improve pretrained generative policies beyond imitation learning. DPPO (Ren et al., 2025) enables online RL for diffusion policies by treating the denoising process as an inner MDP. Extending this paradigm to flow-based policies is less direct, since deterministic ODE sampling provides neither stochastic exploration nor tractable transition likelihoods. In text-to-image generation, Flow-GRPO (Liu et al., 2025) addressed this issue through a marginal-preserving ODE-to-SDE conversion, providing a useful stochasticization principle for flow-based RL. For robotic control, ReinFlow (Zhang et al., 2025b) instead injects learnable noise into discrete deterministic flow transitions to obtain tractable path likelihoods, while π_{RL} (Chen et al., 2025) scales related stochasticization ideas exemplified by Flow-GRPO and ReinFlow to VLA policies through Flow-SDE and Flow-Noise.

As MeanFlow has recently been adopted as a few-step generator across robotic policies and image generation, its RL fine-tuning has attracted increasing attention. DMPO (Zou et al., 2026) transfers ReinFlow-style stochastic transition construction to MeanFlow for robotic control, introducing Gaussian exploration transitions for PPO fine-tuning. MFPO (Dong et al., 2026) instead optimizes MeanFlow policies under maximum-entropy RL through approximate action-likelihood evaluation. In image generation, Flow-Map GRPO (Li et al., 2026) constructs path-preserving stochastic transitions by composing deterministic flow-map transport with anchor-conditioned resampling, enabling GRPO for few-step flow maps such as MeanFlow. Conversely, our method derives tractable Gaussian transitions directly from the underlying flow dynamics, without post-hoc noise injection or anchor-based resampling.

3 Preliminaries

Problem Formulation. We formulate robotic control as a Markov decision process $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P_0, P_{\text{env}}, R, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, P_0 is the initial-state distribution, P_{env} is the environment transition kernel, R is the reward function, and $\gamma \in [0, 1)$ is the discount factor. At environment step k, given $s_k \in \mathcal{S}$, the policy samples action $a_k \sim \pi_\theta(\cdot \mid s_k)$, receives reward $R(s_k, a_k)$, and the environment transitions according to $s_{k+1} \sim P_{\text{env}}(\cdot \mid s_k, a_k)$. The objective is to maximize the expected γ-discounted return $J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{k=0}^{K} \gamma^k R(s_k, a_k)\right]$. The action-value function Q and state-value function V are defined as

$$Q_\theta(s_k, a_k) = \mathbb{E}_{\pi_\theta}\left[\sum_{\tau=k}^{K} \gamma^{\tau-k} R(s_\tau, a_\tau) \mid s_k, a_k\right], \quad V_\theta(s_k) = \mathbb{E}_{a_k \sim \pi_\theta(\cdot|s_k)}[Q_\theta(s_k, a_k)]. \tag{1}$$

The advantage function is defined as $A_\theta(s_k, a_k) := Q_\theta(s_k, a_k) - V_\theta(s_k)$. When estimated from sampled trajectories, we denote the corresponding advantage estimate by $\hat{A}_\theta(s_k, a_k)$.

Flow Matching. Let $x_0 \sim X_0$ be a data sample from the true distribution, and $x_1 \sim X_1$ denote a noise sample ($X_1 = \mathcal{N}(0, I)$ generally). Recent robotic policies (e.g., Black et al., 2024; Physical Intelligence et al., 2025) adopt the rectified flow (Liu et al., 2023) framework, which defines the "noised" data x_τ as $x_\tau = (1-\tau)x_0 + \tau x_1$, for $\tau \in [0, 1]$. Then a model is trained to directly regress the velocity field $v_\theta(x_\tau, \tau)$ by minimizing the Flow Matching objective (Liu et al., 2023):

$$\mathcal{L}(\theta) = \mathbb{E}_{\tau,\, x_0 \sim X_0,\, x_1 \sim X_1}\left[\left\| v - v_\theta(x_\tau, \tau) \right\|^2\right], \tag{2}$$

where the target velocity field is $v = x_1 - x_0$. Over an arbitrary interval $r < t$, the deterministic transport satisfies $x_r = x_t - \int_r^t v_\theta(x_\tau, \tau)\, \mathrm{d}\tau$. Conventional sampling evaluates this integral numerically, typically requiring multiple velocity-field evaluations.

Flow SDE. The deterministic flow dynamics can be converted into a stochastic process that preserves the same marginal distributions while introducing sampling stochasticity (Liu et al., 2025). For rectified flow, the corresponding reverse-time SDE is

$$\mathrm{d}x_\tau = \left[v(x_\tau, \tau) + \frac{\sigma_\tau^2}{2\tau}\big(x_\tau + (1-\tau)v(x_\tau, \tau)\big)\right]\mathrm{d}\tau + \sigma_\tau\, \mathrm{d}\bar{W}_\tau, \tag{3}$$

where $\bar{W}_\tau$ denotes reverse-time Brownian motion and σ_τ controls the stochasticity of the sampling process.

MeanFlow. MeanFlow (Geng et al., 2025) directly models the *average velocity field* over a finite time interval. Given the underlying instantaneous velocity field $v(x_\tau, \tau)$, the average velocity from r to t is defined as

$$u(x_t, r, t) = \frac{1}{t-r}\int_r^t v(x_\tau, \tau)\,\mathrm{d}\tau, \qquad r < t. \tag{4}$$

The instantaneous and average velocity fields satisfy the *MeanFlow Identity*

$$u(x_t, r, t) = v(x_t, t) - (t-r)\frac{\mathrm{d}}{\mathrm{d}t}u(x_t, r, t), \tag{5}$$

where $\frac{\mathrm{d}}{\mathrm{d}t}u(x_t, r, t) = \partial_x u(x_t, r, t)\, v(x_t, t) + \partial_t u(x_t, r, t)$. MeanFlow parameterizes the average velocity with u_θ and trains it through the stop-gradient regression objective

$$\mathcal{L}_{\mathrm{MF}}(\theta) = \mathbb{E}_{r,t,x_t}\left[\left\|u_\theta(x_t, r, t) - \mathrm{sg}(u_{\mathrm{tgt}})\right\|_2^2\right], u_{\mathrm{tgt}} = v(x_t, t) - (t-r)\frac{\mathrm{d}}{\mathrm{d}t}u_\theta(x_t, r, t), \tag{6}$$

where $\mathrm{sg}(\cdot)$ denotes the stop-gradient operator. In practice, the marginal velocity $v(x_t, t)$ can be replaced by its conditional velocity estimator. Given an arbitrary interval $r < t$, the finite-interval sampling update is $x_r = x_t - (t-r)u_\theta(x_t, r, t)$.

4 METHOD

In this section, we first introduce the derivation of **MeanFlow-SDE** (Section 4.1). Then we formulate **MeanFlow Reinforcement Fine-Tuning (MRFT)** by using the MeanFlow-SDE as a stochastic sampling formulation for online exploration and policy optimization (Section 4.2).

4.1 FROM FLOW SDE TO MEANFLOW-SDE: A PRINCIPLED REDUCTION

Our goal is to derive a tractable finite-interval stochastic transition for MeanFlow from the marginal-preserving flow SDE. Consider an arbitrary sampling interval $[r, t]$ with $0 \le r < t < 1$. Integrating Eq. 3 backward from t to r gives

$$x_r = x_t - \int_r^t \frac{\sigma_\tau^2}{2\tau}x_\tau\,\mathrm{d}\tau - \int_r^t \big(1 + \frac{\sigma_\tau^2}{2\tau}(1-\tau)\big)v(x_\tau, \tau)\,\mathrm{d}\tau + \eta_{r,t}, \tag{7}$$

where $\eta_{r,t} = \int_t^r \sigma_\tau\,\mathrm{d}\bar{W}_\tau$ is the accumulated diffusion perturbation. We use $\sigma_\tau = a\sqrt{\frac{\tau}{1-\tau}}$ following (Liu et al., 2025), where a is a scalar hyperparameter that controls the noise level. Since σ_τ is a deterministic scalar diffusion schedule, $\eta_{r,t} \sim \mathcal{N}(0, \Sigma_{r,t}I)$, $\Sigma_{r,t} = \int_r^t \sigma_\tau^2\,\mathrm{d}\tau$.

As Eq. 7 is the exact integral form of the marginal-preserving flow SDE over $[r, t]$, it inherits the marginal-preserving property on this interval. A formal justification is provided in Appendix B. Although the accumulated diffusion term has a tractable Gaussian distribution, the two drift integrals depend on the full stochastic trajectory $\{x_\tau\}_{\tau\in[r,t]}$, preventing direct endpoint-wise evaluation.

We therefore reformulate the drift terms to isolate the velocity integral, enabling its approximation via the MeanFlow field u. Define $g(\tau) = \frac{\sigma_\tau^2}{2\tau}$, $H(\tau) = \int_r^\tau g(s)\,\mathrm{d}s$ and the deterministic temporal weight $W(\tau) = g(\tau)(1-\tau) - H(\tau)$. Substituting the integral representation of x_τ into the first drift term and applying the deterministic and stochastic Fubini theorems yields

$$x_r = (1 - H(t))\,x_t - \int_r^t \big(1 + W(\tau)\big)v(x_\tau, \tau)\,\mathrm{d}\tau + \eta_{r,t} + \mathcal{R}_{r,t}, \tag{8}$$

where $\mathcal{R}_{r,t}$ collects the remaining nested terms and $\mathcal{R}_{r,t} = O\big((t-r)^{3/2-\epsilon}\big)$, almost surely for any $\epsilon \in (0, 1/2)$. The complete derivation is provided in Appendix C. Unless stated otherwise, $O(\cdot)$ below is understood in this pathwise sense. Eq. 8 thus reduces the original drift to a single weighted velocity integral up to a higher-order remainder.

The remaining mismatch is that this integral evaluates v along the stochastic flow SDE trajectory, whereas MeanFlow represents velocity accumulation along the deterministic flow trajectory. Let $\bar{x}_\tau$ satisfy $\mathrm{d}\bar{x}_\tau/\mathrm{d}\tau = v(\bar{x}_\tau, \tau)$ with $\bar{x}_t = x_t$. By the MeanFlow definition, $(t-r)u(x_t, r, t) = \int_r^t v(\bar{x}_\tau, \tau)\,\mathrm{d}\tau$. The following result bounds the discrepancy between the two velocity integrals.

Theorem 1 (MeanFlow Approximation on Stochastic Trajectories)**.** *Suppose that $v(\cdot, \tau)$ is locally Lipschitz in x, uniformly for $\tau \in [r, t]$, and that the flow SDE coefficients satisfy the boundedness conditions stated in Appendix A. Then, for any $\epsilon \in (0, 1/2)$,*

$$\int_r^t v(x_\tau, \tau)\,\mathrm{d}\tau = (t-r)u(x_t, r, t) + O\Big((t-r)^{3/2-\epsilon}\Big). \tag{9}$$

The proof is given in Appendix E. To factor out the temporal weight in Eq. 8, let $m = (r+t)/2$. A midpoint decomposition of W about m (Appendix C) gives

$$\begin{aligned}\int_r^t \big(1 + W(\tau)\big) v(x_\tau, \tau)\,\mathrm{d}\tau &= \big(1 + W(m)\big) \int_r^t v(x_\tau, \tau)\,\mathrm{d}\tau + O\big((t-r)^2\big) \\ &= \big(1 + W(m)\big)(t-r)u(x_t, r, t) + O\Big((t-r)^{3/2-\epsilon}\Big),\end{aligned} \tag{10}$$

for any $\epsilon \in (0, 1/2)$, where the second equality follows from Theorem 1, with the $O((t-r)^2)$ term absorbed into the remainder.

Substituting Eq. 10 into Eq. 8 yields

$$x_r = (1 - H(t))\, x_t - \big(1 + W(m)\big)(t-r)u(x_t, r, t) + \eta_{r,t} + O\Big((t-r)^{3/2-\epsilon}\Big). \tag{11}$$

Hence, Eq. 11 approximates the finite-interval integral form of the marginal-preserving flow SDE over $[r, t]$, with pathwise error $O((t-r)^{3/2-\epsilon})$ for any fixed $\epsilon \in (0, 1/2)$, i.e., with an error exponent arbitrarily close to $3/2$ from below. Dropping the higher-order remainder and replacing u with the learned field u_θ, we define

$$\mu_\theta(x_t, r, t) := (1 - H(t))\, x_t - \big(1 + W(m)\big)(t-r)u_\theta(x_t, r, t). \tag{12}$$

The resulting finite-interval *MeanFlow-SDE* transition is

$$\boxed{x_r = \mu_\theta(x_t, r, t) + \eta_{r,t}, \qquad \eta_{r,t} \sim \mathcal{N}(0, \Sigma_{r,t} I)\,.} \tag{13}$$

Equivalently, the transition kernel is

$$p_\theta(x_r \mid x_t, r, t) = \mathcal{N}(x_r; \mu_\theta(x_t, r, t), \Sigma_{r,t} I)\,. \tag{14}$$

Each MeanFlow-SDE interval therefore requires only one evaluation of u_θ, while providing an explicit Gaussian transition without simulating intermediate stochastic states.

4.2 MeanFlow Reinforcement Fine-Tuning

The MeanFlow-SDE derived in Section 4.1 introduces stochasticity into action generation, enabling exploration during reinforcement fine-tuning. Its Gaussian finite-interval transitions also provide tractable transition likelihoods for policy optimization. Together, these properties enable policy optimization using the tractable joint likelihood of the MeanFlow sampling path, which we instantiate as MeanFlow Reinforcement Fine-Tuning (MRFT). We now make the observation conditioning explicit, which was omitted in Section 4.1 for notational simplicity. Let $0 = t_0 < t_1 < \cdots < t_N = 1$[1] denote the sampling schedule, and let x_k^i be the latent flow state at t_i conditioned on the state s_k. Starting from $x_k^N \sim X_1$, MRFT recursively samples backward according to

$$p_\theta\big(x_k^{i-1} \mid x_k^i, s_k, t_{i-1}, t_i\big) = \mathcal{N}\left(x_k^{i-1}; \mu_\theta(x_k^i, s_k, t_{i-1}, t_i), \Sigma_{t_{i-1}, t_i} I\right), \quad i = N, \ldots, 1, \tag{15}$$

where μ_θ and Σ_{t_{i-1}, t_i} are given by the MeanFlow-SDE transition, with $u_\theta(x_t, r, t)$ replaced by its state-conditioned form $u_\theta(x_t, s_k, r, t)$. The endpoint x_k^0 defines the action a_k executed in the environment.

[1] t_N is truncated in the noise schedule to avoid the endpoint singularity.

Let $\mathbf{x}_k = (x_k^N, \ldots, x_k^0)$ denote the complete sampling path. By the Markov property, its conditional density factorizes as

$$q_\theta(\mathbf{x}_k \mid s_k) = p_1(x_k^N) \prod_{i=1}^{N} p_\theta\big(x_k^{i-1} \mid x_k^i, s_k, t_{i-1}, t_i\big), \tag{16}$$

where p_1 denotes the θ-independent density of X_1. Following the policy-gradient formulation for policies parameterized by a discrete-time Markov process in ReinFlow (Zhang et al., 2025b), the policy gradient can be expressed using the score of the joint sampling-path density, without evaluating the generally intractable marginal likelihood of the executed action. For an episode of horizon K, the resulting policy gradient is

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{k=0}^{K} \gamma^k A_\theta(s_k, a_k) \sum_{i=1}^{N} \nabla_\theta \log p_\theta\big(x_k^{i-1} \mid x_k^i, s_k, t_{i-1}, t_i\big)\right]. \tag{17}$$

Let θ_{old} denote the parameters of the behavior policy used to collect $\mathbf{x}_k$. The path-level likelihood ratio is

$$\rho_k(\theta) = \prod_{i=1}^{N} \frac{p_\theta\big(x_k^{i-1} \mid x_k^i, s_k, t_{i-1}, t_i\big)}{p_{\theta_{\text{old}}}\big(x_k^{i-1} \mid x_k^i, s_k, t_{i-1}, t_i\big)}, \tag{18}$$

which yields the clipped surrogate objective to be minimized,

$$\mathcal{J}_{\text{clip}}(\theta) = \mathbb{E}\left[-\min\Big(\rho_k(\theta)\widehat{A}_{\theta_{old}}(s_k, a_k), \text{clip}\big(\rho_k(\theta), 1-\epsilon_{\text{clip}}, 1+\epsilon_{\text{clip}}\big)\widehat{A}_{\theta_{old}}(s_k, a_k)\Big)\right]. \tag{19}$$

Beyond the on-policy objective above, the same sampling-path likelihood also admits the Q-weighted objective to be minimized,

$$\mathcal{J}_Q(\theta) = \mathbb{E}\left[-Q_{\theta_{old}}(s_k, a_k) \sum_{i=1}^{N} \log p_\theta\big(x_k^{i-1} \mid x_k^i, s_k, t_{i-1}, t_i\big)\right], \qquad a_k = x_k^0, \tag{20}$$

which enables off-policy actor–critic optimization while retaining the same tractable Gaussian transition likelihoods.

5 Experiments

Our experiments are organized around four research questions:

Q1: Few-Step Reinforcement Fine-Tuning. *Can MRFT effectively fine-tune MeanFlow policies under a low-NFE sampling budget?* (Section 5.1)

Q2: Scaling to VLA Policies. *Does MRFT remain effective when scaled to vision-language-action policies?* (Section 5.2)

Q3: Design and Component Analysis. *How do the key design choices in MRFT affect reinforcement fine-tuning performance?* (Section 5.3)

Q4: Real-World Robot Fine-Tuning. *Can MRFT support reinforcement fine-tuning on a real robotic system?* (Section 5.4)

5.1 Comparison with Few-Step Reinforcement Fine-Tuning Methods

Benchmarks. We conduct experiments on nine continuous-control tasks spanning three benchmark suites. OpenAI Gym includes three locomotion tasks: *Ant*, *Hopper*, and *Humanoid*. Franka Kitchen covers three long-horizon manipulation settings: *Complete*, *Mixed*, and *Partial*. Robomimic evaluates visual manipulation on *Can*, *Square*, and *Transport*. Together, these benchmarks cover locomotion, state-based manipulation, and vision-based robotic control.

Baselines. We compare **MRFT** with **DMPO** (Zou et al., 2026), which applies ReinFlow-style noise injection to MeanFlow. We also include **ReinFlow-R** and **ReinFlow-S**, which apply ReinFlow to rectified flow and shortcut flow policies, respectively (Zhang et al., 2025b). All pretrained

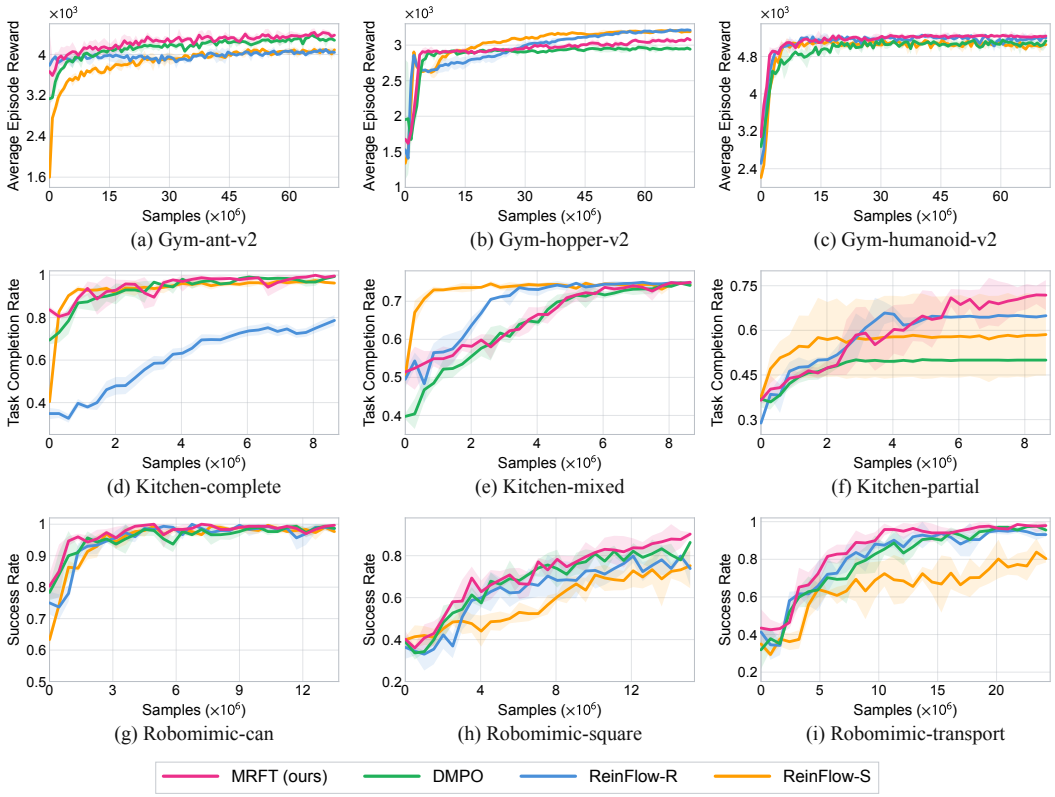


Figure 1: **Training curves on benchmarks.** Solid lines show the mean performance across three independent runs, with shaded regions representing the corresponding standard deviation.

checkpoints are provided by DMPO (Zou et al., 2026); MRFT and DMPO share the same pretrained MeanFlow initialization. For each task, all methods share the same interaction budget, action chunking, and Wasserstein-2 regularization (Zhang et al., 2025b), employing deterministic 2-NFE ODE sampling for evaluation.

To answer **Q1**, Figure 1 compares MRFT with DMPO, ReinFlow-R and ReinFlow-S under a fixed interaction budget. Across the nine tasks, MRFT achieves strong overall performance across all three benchmark suites. Compared with DMPO, which shares the same pretrained MeanFlow initialization, MRFT generally learns faster during early fine-tuning while matching or improving final performance, with larger gains on challenging manipulation tasks. For example, on *Robomimic-can*, MRFT reaches a **95%** success rate with fewer interactions and ultimately achieves **99.7%**, compared with **98.7%** for DMPO. ReinFlow-R and ReinFlow-S remain competitive on locomotion, particularly *Gym-hopper-v2*, and ReinFlow-S learns fastest early on *Kitchen-mixed*, but both are less competitive on the harder visual manipulation tasks (*Robomimic-square* and *Robomimic-transport*). Comprehensive final performance metrics for the nine benchmark tasks are provided in Appendix F.

Table 1: **Final performance and online iteration time on Robomimic.** All methods use the same interaction budget and hardware. SR (%) and WCT (s/itr) denote final success rate and wall-clock time per online iteration, respectively. Results are averaged over three seeds. Parentheses indicate the gap to the column-wise optimum. Higher SR (↑) and lower WCT (↓) are better.

Method	**Can**		**Square**		**Transport**	
	SR↑	**WCT↓**	**SR↑**	**WCT↓**	**SR↑**	**WCT↓**
ReinFlow-R	98.7 (-1.0)	159.3 (+0.1)	73.9 (-16.2)	**172.4**	93.2 (-4.7)	301.5 (+1.7)
ReinFlow-S	97.7 (-2.0)	164.7 (+5.5)	75.2 (-14.9)	194.7 (+22.3)	80.3 (-17.6)	379.6 (+79.8)
DMPO	98.7 (-1.0)	194.8 (+35.6)	86.3 (-3.8)	196.5 (+24.1)	95.5 (-2.4)	340.4 (+40.6)
MRFT (Ours)	**99.7**	**159.2**	**90.1**	178.6 (+6.2)	**97.9**	**299.8**

Table 1 compares final success rates and online iteration times under the same interaction budget and hardware. MRFT achieves the highest mean success rate across all three tasks and the lowest mean iteration time on Can and Transport, being only marginally slower than ReinFlow-R on Square. Compared with DMPO using the same pretrained MeanFlow initialization, MRFT gains 1.0, 3.8, and 2.4 percentage points while reducing iteration time by 18.3%, 9.1%, and 11.9% on Can, Square, and Transport, respectively.

5.2 Evaluation on $\pi_{0.5}$-Style Vision-Language-Action Policies

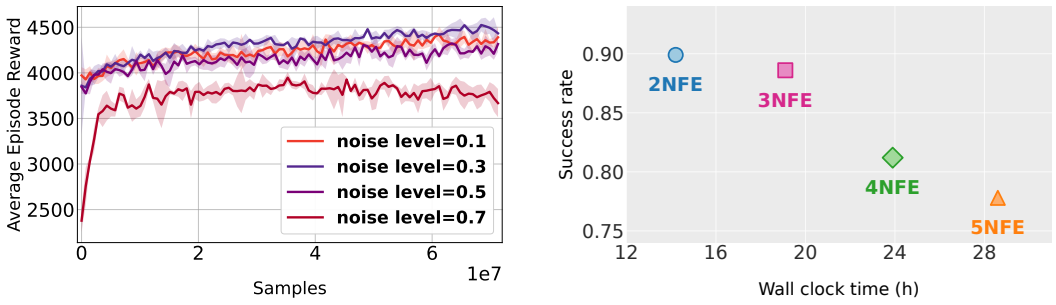
To address **Q2**, we evaluate MRFT with a $\pi_{0.5}$-style VLA policy on the four LIBERO suites. We adopt the LIBERO benchmark configurations detailed in π_{RL} (Chen et al., 2025), strictly following their policy inputs, demonstration budgets, and evaluation protocols. The pretrained $\pi_{0.5}$ vision-language backbone remains frozen throughout SFT and RL, while the action expert is initialized from scratch for SFT and trained with the improved MeanFlow objective (Geng et al., 2026). We consider full-data SFT and few-shot SFT, where the latter uses one demonstration per task (40 trajectories in total) and serves as the initialization for RL fine-tuning. We compare MRFT against a Flow-Noise baseline adapted from π_{RL}. Flow-Noise retains the MeanFlow sampling dynamics while learning the injected noise scale with a neural network. Both methods start from the same few-shot MeanFlow SFT checkpoint and use the same RL configuration.

As shown in Table 2, MRFT improves the average success rate from 90.4% to 97.6%, a gain of 7.2 percentage points over the few-shot SFT initialization. It outperforms Flow-Noise on all four LIBERO suites, increasing the average from 96.3% to 97.6%, and also surpasses the full-data SFT reference of 95.4%. These results answer **Q2** affirmatively, showing that MRFT remains effective at VLA scale, as exemplified by the $\pi_{0.5}$-style architecture.

Table 2: **Evaluation on LIBERO with a $\pi_{0.5}$-style VLA policy.** Success rates (%) are reported. Flow-Noise and MRFT share the same few-shot MeanFlow SFT initialization and use 2-NFE for RL rollouts and evaluation. Δ Avg. is the absolute gain over few-shot SFT.

Model		LIBERO					
		Spatial	Object	Goal	Long	Avg.	Δ Avg.
# Full-Data SFT							
$\pi_{0.5}$-MeanFlow	SFT	98.0	95.4	96.8	91.4	95.4	–
# Few-Shot SFT + RL							
$\pi_{0.5}$-MeanFlow	SFT	97.4	83.8	91.0	89.4	90.4	–
	Flow-Noise	98.8	93.6	98.2	94.4	96.3	+5.9
	MRFT	**99.0**	**97.6**	**98.8**	**95.0**	**97.6**	**+7.2**

5.3 Ablation Studies and Design Choices in MRFT

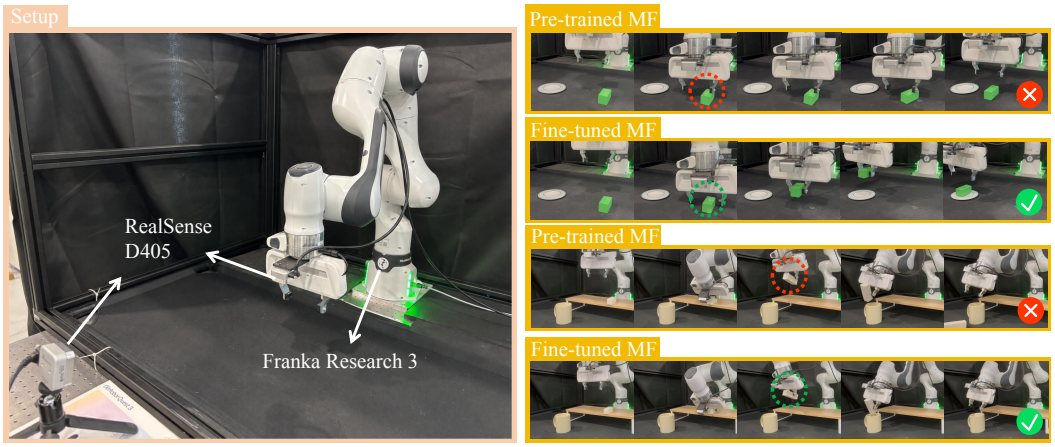


(a) Effect of noise level on Gym-ant-v2

(b) Effect of sampling steps on Robomimic-square

Figure 2: **Ablation analysis of key design components.**

This section addresses **Q3** by analyzing the effects of the noise level and sampling steps. In MeanFlow-SDE, $\sigma_\tau = a\sqrt{\frac{\tau}{1-\tau}}$, where the noise level a controls exploration strength. As shown



(a) Settings of real world (b) Pre-trained vs. Fine-tuned MeanFlow policies (MF)

Figure 3: **Real-world evaluation.** (a) Environmental setup. (b) Representative rollouts on two pick-and-place tasks before and after MeanFlow reinforcement fine-tuning.

in Figure 2a, performance peaks at $a = 0.3$: weaker noise limits exploration, while larger a progressively degrades performance, with a pronounced drop at $a = 0.7$. This suggests that moderate stochasticity provides the best balance between exploration and excessive action perturbation. We further study the effect of sampling steps on RoboMimic-square. In Figure 2b, 2-NFE and 3-NFE achieve comparable performance, while 4-NFE and 5-NFE lead to lower success rates and higher end-to-end wall-clock cost for the RL fine-tuning pipeline. This confirms that more sampling steps do not necessarily benefit RL fine-tuning of MeanFlow policies. We therefore use 2-NFE by default.

5.4 Reinforcement Fine-Tuning on a Real Robot System

To answer **Q4**, we evaluate MRFT on two real-world manipulation tasks using a Franka Research 3 robot. The robot is equipped with wrist- and agent-view Intel RealSense D405 cameras, as shown in Figure 3a. Specifically, we use a SpaceMouse to collect 10 demonstrations for *Cube-to-Saucer Placement* and 20 for *Cube-into-Cup Placement*, initializing the MeanFlow policies via the Flow-T architecture (Zhang et al., 2026). Starting from the demonstration-trained policies, we perform online reinforcement fine-tuning directly on the physical robot. We instantiate the off-policy formulation of MRFT with SAC (Haarnoja et al., 2018), using 2-NFE MeanFlow-SDE sampling for action generation.

Figure 3b shows representative rollouts before and after reinforcement fine-tuning. Evaluation is conducted with small variations in the initial object pose relative to the demonstration configurations. The pre-trained MeanFlow policies can fail under these variations, whereas the fine-tuned policies successfully complete the grasp-and-place sequence. MRFT improves the success rate from 25% to 60% on *Cube-to-Saucer Placement* and from 45% to 70% on *Cube-into-Cup Placement*, over 20 trials per task. These results show that MRFT can improve demonstration-initialized MeanFlow policies through online reinforcement learning on physical hardware.

6 Conclusion

Standard MeanFlow policies preclude direct reinforcement fine-tuning, as their deterministic nature lacks the stochastic exploration and tractable likelihoods RL requires. We resolve this by deriving MeanFlow-SDE, leveraging the underlying flow dynamics to natively establish finite-interval stochastic transitions. This theoretical grounding yields MRFT, a principled MeanFlow fine-tuning framework without ad-hoc auxiliary noise modules. Our evaluations demonstrate MRFT can enhance performance and accelerate training speed across many robotic tasks. Future work will explore tighter residual bounds for reliable 1-NFE optimization, and extend MRFT to more complex robotic systems.

AI USE STATEMENT

Generative AI tools were used in this work exclusively as general-purpose writing aids for light language polishing, including minor corrections to grammar, phrasing, and style. They were not employed in any core research capacity. Specifically, no AI tools were used to generate datasets, develop theoretical frameworks, formulate mathematical claims, design experiments, implement code, or analyze data. All research ideas, problem formulations, methodologies, and substantive manuscript drafting were conducted solely by the authors. We have reviewed and verified all AI-assisted language edits and assume full responsibility for the final content, claims, and artifacts presented in this work. No language model qualifies as an author or intellectual contributor.

REPRODUCIBILITY STATEMENT

To support reproducibility, we provide source code and demonstration videos in the supplementary materials to facilitate replication and inspection of the reported results.

REFERENCES

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Kang Chen, Zhihao Liu, Tonghe Zhang, Zhen Guo, Si Xu, Hao Lin, Hongzhi Zang, Quanlu Zhang, Zhaofei Yu, Guoliang Fan, et al. π_{RL}: Online rl fine-tuning for flow-based vision-language-action models. *arXiv preprint arXiv:2510.25889*, 2025.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems XIX, Daegu, Republic of Korea, July 10-14, 2023*, 2023.

Eugenio Chisari, Nick Heppert, Max Argus, Tim Welschehold, Thomas Brox, and Abhinav Valada. Learning robotic manipulation policies from point clouds with conditional flow matching. In *Conference on Robot Learning, 6-9 November 2024, Munich, Germany*, 2024.

Xiaoyi Dong, Xi Sheryl Zhang, and Jian Cheng. Mean flow policy optimization. *arXiv preprint arXiv:2604.14698*, 2026.

Han Fang, Yize Huang, Yuheng Zhao, Paul Weng, Xiao Li, and Yutong Ban. OMP: one-step mean-flow policy with directional alignment. *arXiv preprint arXiv:2512.19347*, 2025.

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*, 2025.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025.

Zhengyang Geng, Yiyang Lu, Zongze Wu, Eli Shechtman, J. Zico Kolter, and Kaiming He. Improved mean flows: On the challenges of fastforward generative models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2026, Denver, CO, USA, June 3-7, 2026*, 2026.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, 2018.

Xixi Hu, Qiang Liu, Xingchao Liu, and Bo Liu. Adaflow: Imitation learning with variance-adaptive flow-based policies. In *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

Zhiqi Li, Wen Zhang, and Bo Zhu. Flow-map GRPO: reinforcement learning for few-step flow-map generators via anchored stochastic composition. *arXiv preprint arXiv:2607.00535*, 2026.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, 2023.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online RL. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, 2023.

Seohong Park, Qiyang Li, and Sergey Levine. Flow q-learning. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, 2025.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Aaditya Prasad, Kevin Lin, Jimmy Wu, Linqi Zhou, and Jeannette Bohg. Consistency policy: Accelerated visuomotor policies via consistency distillation. In *Robotics: Science and Systems XX, Delft, The Netherlands, July 15-19, 2024*, 2024.

Allen Z. Ren, Justin Lidard, Lars Lien Ankile, Anthony Simeonov, Pulkit Agrawal, Anirudha Majumdar, Benjamin Burchfiel, Hongkai Dai, and Max Simchowitz. Diffusion policy policy optimization. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*, 2025.

Juyi Sheng, Ziyi Wang, Peiming Li, and Mengyuan Liu. MP1: meanflow tames policy learning in 1-step for robotic manipulation. In *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, 2026.

Andreas Sochopoulos, Nikolay Malkin, Nikolaos Tsagkas, João Moura, Michael Gienger, and Sethu Vijayakumar. Fast flow-based visuomotor policies via conditional optimal transport couplings. *arXiv preprint arXiv:2505.01179*, 2025.

Zhendong Wang, Max Li, Ajay Mandlekar, Zhenjia Xu, Jiaojiao Fan, Yashraj Narang, Linxi Fan, Yuke Zhu, Yogesh Balaji, Mingyuan Zhou, Ming-Yu Liu, and Yu Zeng. One-step diffusion policy: Fast visuomotor policies via diffusion distillation. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, 2025.

Ge Yan, Jiyue Zhu, Yuquan Deng, Shiqi Yang, Ri-Zhao Qiu, Xuxin Cheng, Marius Memmel, Ranjay Krishna, Ankit Goyal, Xiaolong Wang, and Dieter Fox. Maniflow: A general robot manipulation policy via consistency flow training. *arXiv preprint arXiv:2509.01819*, 2025.

Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. In *Robotics: Science and Systems XX, Delft, The Netherlands, July 15-19, 2024*, 2024.

Guojian Zhan, Letian Tao, Pengcheng Wang, Yixiao Wang, Yuxin Chen, Yiheng Li, Hongyang Li, Masayoshi Tomizuka, and Shengbo Li. Mean flow policy with instantaneous velocity constraint for one-step action generation. In *International Conference on Learning Representations*, volume 2026, pp. 157132–157154, 2026.

Qinglun Zhang, Zhen Liu, Haoqiang Fan, Guanghui Liu, Bing Zeng, and Shuaicheng Liu. Flowpolicy: Enabling fast and robust 3d flow-based policy via consistency flow matching for robot manipulation. In *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, 2025a.

Tonghe Zhang, Chao Yu, Sichang Su, and Yu Wang. Reinflow: Fine-tuning flow matching policy with online reinforcement learning. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025b.

Yixian Zhang, Shu-ang Yu, Tonghe Zhang, Mo Guang, Haojia Hui, Kaiwen Long, Yu Wang, Chao Yu, and Wenbo Ding. Sac flow: Sample-efficient reinforcement learning of flow-based policies via velocity-reparameterized sequential modeling. In *International Conference on Learning Representations*, volume 2026, pp. 121983–122016, 2026.

Guowei Zou, Haitao Wang, Hejun Wu, Yukun Qian, Yuhang Wang, and Weibing Li. DM1: meanflow with dispersive regularization for 1-step robotic manipulation. *arXiv preprint arXiv:2510.07865*, 2025.

Guowei Zou, Haitao Wang, Hejun Wu, Yukun Qian, Yuhang Wang, and Weibing Li. One step is enough: Dispersive meanflow policy optimization. *arXiv preprint arXiv:2601.20701*, 2026.

Appendix Overview

This appendix provides supplementary material organized as follows:

Appendix A: Standing Assumptions and Notation states the regularity, well-posedness, and notation used throughout the theoretical analysis.

Appendix B: Marginal Preservation of the Finite-Interval Flow-SDE shows that its exact finite-interval integral form inherits the known marginal path.

Appendix C: Reduction of the State-Dependent Drift Integral derives the weighted velocity integral and bounds the remaining state-dependent and stochastic terms.

Appendix D: Midpoint Reduction of the Weighted Velocity Integral proves the pathwise $O((t-r)^2)$ error incurred by evaluating the temporal weight at the interval midpoint.

Appendix E: MeanFlow Approximation of the Stochastic Velocity Integral proves the stochastic-to-deterministic velocity approximation.

Appendix F: Comprehensive Final Performance Metrics reports detailed final results across Gym locomotion, Franka Kitchen, and RoboMimic, with comparisons to all reinforcement fine-tuning baselines.

A Standing Assumptions and Notation

We work on $[r,t] \subset I$, where $I \subset [0,1)$ is a fixed compact interval, and set $h = t - r$ and $m = (r+t)/2$. The restriction away from $t = 1$ keeps the coefficients bounded for the noise schedule used in Section 4.1. The following assumptions apply to the exact velocity v and the continuous dynamics.

Assumption 1 (Regularity of the velocity field)**.** *The exact velocity field $v : \mathbb{R}^d \times I \to \mathbb{R}^d$ is jointly continuous and locally Lipschitz in the state uniformly over I. Specifically, for every compact set $K \subset \mathbb{R}^d$, there exists $L_K < \infty$ such that*

$$\|v(x,\tau) - v(y,\tau)\| \le L_K \|x - y\|, \qquad x, y \in K,\ \tau \in I.$$

Moreover, v has at most linear growth:

$$\|v(x,\tau)\| \le C_v(1 + \|x\|), \qquad x \in \mathbb{R}^d,\ \tau \in I,$$

for some constant $C_v < \infty$.

Assumption 2 (Regularity of the diffusion schedule)**.** *The diffusion schedule σ_τ is deterministic, nonnegative, and bounded on I. The function g defined in Eq. A.1 is Lipschitz continuous on I.*

Assumption 3 (Well-posedness and moment conditions)**.** *The deterministic and stochastic dynamics are well posed on I, with continuous sample paths and finite second moments.*

Pathwise order and common notation. Fix $I = [\underline{\tau}, \overline{\tau}] \subset [0,1)$ and construct one exact reverse-time SDE trajectory by drawing $x_{\overline{\tau}} \sim p_{\overline{\tau}}$ independently of its Brownian driver. Every interval $[r,t] \subset I$ below is a restriction of this same trajectory; the deterministic comparison trajectory starts from its state x_t. Throughout this appendix, $X_{r,t} = O(h^\alpha)$ means a pathwise almost-sure estimate as $h = t - r \to 0$: on an event of probability one, $\|X_{r,t}\| \le Ch^\alpha$ for all sufficiently small $[r,t] \subset I$, with a finite random constant C independent of the chosen subinterval. Constants may depend on the sample path and I. When the exponent contains ϵ, it is fixed before taking $h \to 0$, and the constant and probability-one event may depend on it. As in Section 4.1, write

$$g(\tau) = \frac{\sigma_\tau^2}{2\tau}, \qquad H(\tau) = \int_r^\tau g(s)\,\mathrm{d}s, \qquad W(\tau) = g(\tau)(1-\tau) - H(\tau). \tag{A.1}$$

The field u is the exact average velocity along the deterministic flow initialized at x_t.

For the schedule $\sigma_\tau = a\sqrt{\tau/(1-\tau)}$ used in the main text, these quantities and the diffusion variance have the closed forms

$$g(\tau) = \frac{a^2}{2(1-\tau)}, \qquad H(\tau) = \frac{a^2}{2}\log\frac{1-r}{1-\tau},$$
$$W(\tau) = \frac{a^2}{2}\left(1 - \log\frac{1-r}{1-\tau}\right), \qquad \Sigma_{r,t} = a^2\left(\log\frac{1-r}{1-t} - h\right). \tag{A.2}$$

Here $a > 0$; the formulas are finite for $0 \le r < t < 1$.

B Marginal Preservation of the Finite-Interval Flow-SDE

Let p_τ be the density of the deterministic-flow marginal in Section 3. For the exact velocity v, the reverse-time Flow-SDE in Eq. 3 preserves p_τ (Liu et al., 2025). We show that its *exact* finite-interval integral form inherits this property.

Write the SDE drift as

$$b(x,\tau) = v(x,\tau) + g(\tau)\big(x + (1-\tau)v(x,\tau)\big). \tag{B.1}$$

For a fixed t, use the forward clock $u = t - \tau$ and denote its Brownian driver by B_u, independent of x_t. Set

$$\eta_{\tau,t} := \int_t^\tau \sigma_s \,\mathrm{d}\bar{W}_s = \int_0^{t-\tau} \sigma_{t-u}\,\mathrm{d}B_u, \qquad \Sigma_{\tau,t} := \int_\tau^t \sigma_s^2\,\mathrm{d}s. \tag{B.2}$$

Since σ is deterministic, $\eta_{\tau,t} \sim \mathcal{N}(0, \Sigma_{\tau,t} I_d)$, where I_d is the d-dimensional identity.

Proposition 1 (Exact finite-interval propagation)**.** *Under the standing assumptions, the integral family*

$$x_\tau = x_t - \int_\tau^t b(x_s, s)\,\mathrm{d}s + \eta_{\tau,t}, \qquad \tau \in [r,t], \tag{B.3}$$

is equivalent to Eq. 3 on $[r,t]$. Hence, by the cited marginal preservation result, if $x_t \sim p_t$, then $x_\tau \sim p_\tau$ for every $\tau \in [r,t]$. In particular, the exact interval kernel $K_{r,t}(x,A) := \mathbb{P}(x_r \in A \mid x_t = x)$ satisfies

$$\int_{\mathbb{R}^d} K_{r,t}(x,A) p_t(x)\,\mathrm{d}x = \int_A p_r(y)\,\mathrm{d}y \tag{B.4}$$

for every measurable set A.

Proof. Set $Y_u = x_{t-u}$. In the forward clock, Eq. 3 is the Itô equation

$$Y_u = x_t - \int_0^u b(Y_s, t-s)\,\mathrm{d}s + \int_0^u \sigma_{t-s}\,\mathrm{d}B_s. \tag{B.5}$$

Substituting $u = t - \tau$ gives Eq. B.3. Conversely, that family of identities gives Eq. B.5. By well-posedness, both forms have the same adapted solution. The cited result gives $x_\tau \sim p_\tau$. Conditioning on x_t gives the left-hand side of Eq. B.4; $x_r \sim p_r$ gives its right-hand side. □

At $\tau = r$, expanding b in Eq. B.3 gives the exact identity used in Section 4.1:

$$x_r = x_t - \int_r^t g(\tau)x_\tau\,\mathrm{d}\tau - \int_r^t \big(1 + g(\tau)(1-\tau)\big)v(x_\tau,\tau)\,\mathrm{d}\tau + \eta_{r,t}. \tag{B.6}$$

This identity retains the entire intermediate path. Its exact marginal property does not transfer automatically to a transition obtained by approximating the drift integrals.

C Reduction of the State-Dependent Drift Integral

C.1 Eliminating the state-dependent drift

For $\tau \in [r, t]$, apply Eq. B.6 on the shorter interval $[\tau, t]$:

$$x_\tau = x_t - \int_\tau^t g(s) x_s \, \mathrm{d}s - \int_\tau^t \big(1 + g(s)(1-s)\big) v(x_s, s) \, \mathrm{d}s + \eta_{\tau,t}. \tag{C.1}$$

The state integral in Eq. B.6 can now be reduced without replacing x_τ by an endpoint. The triangular-domain identity

$$\int_r^t g(\tau) \int_\tau^t F(s) \, \mathrm{d}s \, \mathrm{d}\tau = \int_r^t H(s) F(s) \, \mathrm{d}s \tag{C.2}$$

holds for either drift integrand in Eq. C.1. Likewise, stochastic Fubini gives

$$\int_r^t g(\tau) \eta_{\tau,t} \, \mathrm{d}\tau = \int_t^r H(s) \sigma_s \, \mathrm{d}\bar{W}_s. \tag{C.3}$$

In the forward clock fixed above, both sides equal $\int_0^h H(t-s)\sigma_{t-s} \, \mathrm{d}B_s$. The interchange is valid because g and σ are bounded and deterministic on I. Substitution therefore yields

$$\int_r^t g(\tau) x_\tau \, \mathrm{d}\tau = H(t) x_t - \int_r^t H(\tau) g(\tau) x_\tau \, \mathrm{d}\tau - \int_r^t H(\tau) \big(1 + g(\tau)(1-\tau)\big) v(x_\tau, \tau) \, \mathrm{d}\tau + \int_r^t g(\tau) \eta_{\tau,t} \, \mathrm{d}\tau. \tag{C.4}$$

Inserting this expression into Eq. B.6 and collecting the velocity terms by $W(\tau) = g(\tau)(1-\tau) - H(\tau)$ gives the exact identity

$$x_r = \big(1 - H(t)\big) x_t - \int_r^t \big(1 + W(\tau)\big) v(x_\tau, \tau) \, \mathrm{d}\tau + \eta_{r,t} + \mathcal{R}_{r,t}, \tag{C.5}$$

where the residual is

$$\mathcal{R}_{r,t} = \int_r^t H(\tau) g(\tau) \big(x_\tau + (1-\tau) v(x_\tau, \tau)\big) \, \mathrm{d}\tau - \int_r^t g(\tau) \eta_{\tau,t} \, \mathrm{d}\tau. \tag{C.6}$$

Eq. C.5 and Eq. C.6 are the reduction and residual used in Eq. 8 of the main text.

C.2 Pathwise size of the residual

Set $\bar{g} = \sup_{\tau \in I} |g(\tau)|$. Continuity of the trajectories and v gives an almost surely finite pathwise bound $B = \sup_{\tau \in I} \|x_\tau + (1-\tau) v(x_\tau, \tau)\|$. Since $|H(\tau)| \le \bar{g}(\tau - r)$, the first term of Eq. C.6 is bounded by $\bar{g}^2 B h^2 / 2$.

For every moment order $\ell \ge 2$, the bounded deterministic diffusion coefficient gives $\mathbb{E}\|\eta_{s,t}\|^\ell \le C_\ell |t-s|^{\ell/2}$ on I. Kolmogorov's continuity theorem therefore gives almost surely Hölder-continuous noise paths of every order below $1/2$. For each $\epsilon \in (0, 1/2)$, on the common coupling specified in Appendix A, there is an almost surely finite C_ϵ such that

$$\|\eta_{\tau,t}\| \le C_\epsilon (t - \tau)^{1/2 - \epsilon}, \qquad r \le \tau \le t. \tag{C.7}$$

Consequently,

$$\|\mathcal{R}_{r,t}\| \le \frac{\bar{g}^2 B}{2} h^2 + \frac{\bar{g} C_\epsilon}{3/2 - \epsilon} h^{3/2 - \epsilon} = O(h^{3/2-\epsilon}). \tag{C.8}$$

The h^2 term comes from the nested drift; the $h^{3/2-\epsilon}$ term comes from integrated Brownian increments. The weighted velocity integral in Eq. C.5 has the midpoint decomposition used in Section 4.1:

$$\int_r^t \big(1 + W(\tau)\big) v(x_\tau, \tau) \, \mathrm{d}\tau = \big(1 + W(m)\big) \int_r^t v(x_\tau, \tau) \, \mathrm{d}\tau + \mathcal{E}_{r,t}^{\mathrm{mid}}, \qquad \mathcal{E}_{r,t}^{\mathrm{mid}} = O(h^2). \tag{C.9}$$

Appendix D proves the stated bound.

D Midpoint Reduction of the Weighted Velocity Integral

The midpoint step in Eq. C.9 is proved here. Let L_g be a Lipschitz constant for g on I. By Eq. A.1, W is Lipschitz with, for example, $L_W = L_g + 2\bar{g}$. The exact midpoint error is

$$\mathcal{E}_{r,t}^{\mathrm{mid}} = \int_r^t \big(W(\tau) - W(m)\big) v(x_\tau, \tau)\, \mathrm{d}\tau. \tag{D.1}$$

Along the stochastic trajectory, $M = \sup_{\tau \in I} \|v(x_\tau, \tau)\| < \infty$ almost surely. Thus

$$\|\mathcal{E}_{r,t}^{\mathrm{mid}}\| \le L_W M \int_r^t |\tau - m|\, \mathrm{d}\tau = \frac{L_W M}{4} h^2. \tag{D.2}$$

For the schedule in Eq. A.2, $W'(\tau) = -a^2/[2(1-\tau)]$, so L_W is uniform over intervals contained in I.

E MeanFlow Approximation of the Stochastic Velocity Integral

Let $\bar{x}_\tau$ be the deterministic flow with the same terminal state as the stochastic process:

$$\frac{\mathrm{d}\bar{x}_\tau}{\mathrm{d}\tau} = v(\bar{x}_\tau, \tau), \qquad \bar{x}_t = x_t. \tag{E.1}$$

The exact MeanFlow definition in Eq. 4 therefore reads

$$h\, u(x_t, r, t) = \int_r^t v(\bar{x}_\tau, \tau)\, \mathrm{d}\tau. \tag{E.2}$$

The stochastic and deterministic velocity integrals differ because their trajectories differ. Set $\delta_\tau = x_\tau - \bar{x}_\tau$. Subtracting their integral equations yields

$$\delta_\tau = -\int_\tau^t \big(v(x_s, s) - v(\bar{x}_s, s)\big)\, \mathrm{d}s - \int_\tau^t g(s)\big(x_s + (1-s)v(x_s, s)\big)\, \mathrm{d}s + \eta_{\tau,t}. \tag{E.3}$$

The stochastic path is bounded on I almost surely. Linear growth of v and Grönwall's inequality also bound every deterministic comparison path $\bar{x}_\tau$ initialized at x_t, uniformly over $r \le \tau \le t$ and $[r, t] \subset I$. Thus both families lie in one path-dependent compact ball. Let L_v be a spatial Lipschitz constant for v on that ball, and let B be the pathwise bound defined in Appendix C.2. Both are independent of h. Equation C.7 then implies

$$\|\delta_\tau\| \le L_v \int_\tau^t \|\delta_s\|\, \mathrm{d}s + \bar{g}B(t-\tau) + C_\epsilon (t-\tau)^{1/2-\epsilon}. \tag{E.4}$$

The integral form of Grönwall's inequality gives

$$\|\delta_\tau\| \le e^{L_v h}\left[\bar{g}B(t-\tau) + C_\epsilon (t-\tau)^{1/2-\epsilon}\right]. \tag{E.5}$$

Using Eq. E.2 and integrating this bound once more,

$$\begin{aligned}
&\left\| \int_r^t v(x_\tau, \tau)\, \mathrm{d}\tau - h\, u(x_t, r, t) \right\| \\
&\quad \le L_v \int_r^t \|\delta_\tau\|\, \mathrm{d}\tau \\
&\quad \le L_v e^{L_v h}\left[\frac{\bar{g}B}{2} h^2 + \frac{C_\epsilon}{3/2-\epsilon} h^{3/2-\epsilon}\right] \\
&\quad = O(h^{3/2-\epsilon}).
\end{aligned} \tag{E.6}$$

Consequently, for every $\epsilon \in (0, 1/2)$,

$$\int_r^t v(x_\tau, \tau)\, \mathrm{d}\tau = h\, u(x_t, r, t) + O(h^{3/2-\epsilon}), \tag{E.7}$$

which proves Theorem 1.

F Comprehensive Final Performance Metrics

Table 3 reports the final performance of all methods across Gym locomotion, Franka Kitchen, and Robomimic, with mean and standard deviation computed over three independent seeds. Across the nine tasks, MRFT achieves the highest mean performance on eight, including all six manipulation tasks. The only exception is Gym-hopper-v2, where ReinFlow-R performs best. These results complement the learning curves in the main paper and show that the advantage of MRFT is retained at convergence across substantially different control domains.

Gym locomotion. MRFT achieves the best final return on both Ant and Humanoid. On Gym-Ant-v2, it reaches 4376 ± 33.9, compared with 4277 ± 39.6 for the strongest baseline, DMPO, corresponding to an improvement of 99 return points (2.3%). On Gym-Humanoid-v2, MRFT obtains 5238 ± 13.3, exceeding the 5198 ± 155.3 of ReinFlow-R while exhibiting substantially lower variance across seeds. The trend is different on Gym-Hopper-v2: MRFT obtains 3074 ± 31.9, below ReinFlow-R (3208 ± 16.1) and ReinFlow-S (3188 ± 15.3), but above DMPO (2942 ± 25.2). Hence, the gains on locomotion are task dependent, with MRFT providing the strongest final performance on the higher-dimensional Ant and Humanoid tasks.

Franka Kitchen. The advantage of MRFT is more consistent on long-horizon manipulation. MRFT achieves the best completion rate on all three Kitchen tasks: $99.6 \pm 0.3\%$ on Kitchen-complete, $75.0 \pm 0.0\%$ on Kitchen-mixed, and $71.8 \pm 5.1\%$ on Kitchen-partial. Relative to the strongest competing result on each task, these correspond to absolute improvements of 0.3, 0.2, and 6.9 percentage points, respectively. The improvement is particularly pronounced on Kitchen-partial, where the best baseline reaches 64.9%, suggesting that MRFT remains effective when task completion is more challenging and the pretrained policy leaves greater room for reinforcement fine-tuning.

Robomimic. MRFT also consistently improves final success rates on visual manipulation. It achieves $99.7 \pm 0.6\%$ on Can, $90.1 \pm 0.9\%$ on Square, and $97.9 \pm 1.9\%$ on Transport, outperforming the strongest baseline means of 98.7%, 86.3%, and 95.5%, respectively. This yields absolute gains of 1.0, 3.8, and 2.4 percentage points. The improvement on Square is especially notable: MRFT not only raises the success rate from 86.3% for DMPO to 90.1%, but also reduces the standard deviation from 1.7 to 0.9. Together with the near-saturated performance on Can and Transport, these results show that the improvements persist across manipulation tasks with different levels of difficulty.

Overall. Taken together, MRFT obtains the best final mean performance on eight of nine tasks and on every manipulation task considered. While the locomotion results show that no single method dominates uniformly on all environments, the consistent improvements across Franka Kitchen and RoboMimic indicate that MRFT is particularly effective for the manipulation settings targeted in this work.

Table 3: Final performance metrics. Mean $\pm$ Std over 3 seeds. **Bold** = best, underlined = 2nd-best.

Task	ReinFlow-R	ReinFlow-S	DMPO	MRFT (ours)
Gym Locomotion Tasks (Average Episode Reward)				
Gym-ant-v2	4040 ± 82.4	4090 ± 14.2	4277 ± 39.6	**4376 ± 33.9**
Gym-hopper-v2	**3208 ± 16.1**	3188 ± 15.3	2942 ± 25.2	3074 ± 31.9
Gym-humanoid-v2	5198 ± 155.3	5054 ± 49.1	5134 ± 134.1	**5238 ± 13.3**
Franka Kitchen Tasks (Task Completion Rate, %)				
Kitchen-complete	78.7 ± 2.5	96.3 ± 0.7	99.3 ± 0.6	**99.6 ± 0.3**
Kitchen-mixed	74.8 ± 0.1	74.4 ± 0.6	74.2 ± 0.3	**75.0 ± 0.0**
Kitchen-partial	64.9 ± 0.1	58.6 ± 13.8	50.0 ± 0.0	**71.8 ± 5.1**
Robomimic Tasks (Success Rate, %)				
Robomimic-can	98.7 ± 1.5	97.7 ± 0.6	98.7 ± 1.5	**99.7 ± 0.6**
Robomimic-square	73.9 ± 8.3	75.2 ± 7.8	86.3 ± 1.7	**90.1 ± 0.9**
Robomimic-transport	93.2 ± 1.0	80.3 ± 2.1	95.5 ± 3.6	**97.9 ± 1.9**