ReFPO: Reflow Regularization for Flow Matching Policy Gradients

Ge Wang[1,2,3] **Yibo Peng**[1,2,3] **Fan Feng**[2,3] **Shenhao Yan**[2,3]
Chengsi Yao[2,3] **Jiahao Yang**[2,3] **Honghao Cai**[1,2,3] **Yiming Zhao**[3,1,2]
Xi Li[3] **Jinke Ren**[1,2] **Shuguang Cui**[1,2]
Yatong Han[3,1,2,†] **Zhen Li**[2,1,†]

[1]**CUHK-Shenzhen** [2]**FNii-Shenzhen** [3]**Ising AI**

Abstract

We present Reflow-regularized Flow Matching Policy Gradients (ReFPO), an online reinforcement learning method that improves one-step flow policies with an advantage-independent conditional flow matching (CFM) term. Our analysis identifies a signed local update on the active surrogate branch of Flow Policy Optimization (FPO): positive-advantage samples receive straight-target fitting, whereas negative-advantage samples receive the opposite update. This motivates a Reflow-style regularizer on policy-generated actions, with an explicit distinction from classical transport-pair Reflow. ReFPO reuses the existing CFM residual without an additional flow-model forward pass or a separate distillation stage. On ten MuJoCo Playground tasks, average one-step return increases from 565 for noise-supervised FPO to 690 for velocity-supervised ReFPO. A controlled FPO++ comparison that changes only the regularizer improves one-step return by 10.7%. GridWorld illustrates multimodal control, and humanoid experiments demonstrate effective one-step deployment. In a one-iteration, five-paired-seed BallInCup intervention, adding the negative-advantage regularizer reduces held-out CFM-error growth on positive-advantage actions, consistent with local protection of the shared velocity field. The code is available at *ReFPO*.

1 Introduction

Generative policies can represent multimodal actions [4], but iterative sampling increases latency. Rectified Flow, consistency models, and distillation enable faster generation [11, 19, 26]. We ask how online policy optimization can improve one-step control.

Flow Policy Optimization (FPO) [15] replaces the intractable likelihood ratio of an implicit flow policy with an exponential difference of conditional flow matching (CFM) losses. This creates a specific connection between reward optimization and straight-target velocity fitting. For a frozen sample on the active surrogate branch, the gradient coefficient of the CFM loss is Ar: positive advantage yields CFM descent, while negative advantage yields anti-CFM ascent. The action is generated by the behavior policy, but the noise used for the CFM proxy is sampled afresh. We therefore interpret the connection as signed, local Reflow-style fitting on the self-generated action marginal, with its precise scope given in Section 3.2.

ReFPO adds an unweighted CFM term to the FPO objective (Figure 1). This shifts the active-branch coefficient from Ar to $Ar + \lambda$, opposing CFM ascent on negative-advantage samples. On clipped plateaus, the added term continues to fit the target. ReFPO reuses the residual that FPO already computes for the proxy ratio, so the regularizer requires no additional flow-model forward pass.

[†]Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

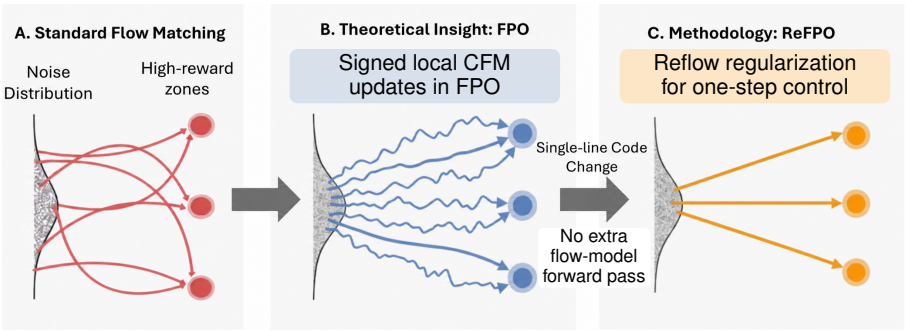


Figure 1: **Overview of ReFPO.** On FPO's active branch, positive advantages favor CFM fitting and negative advantages oppose it. ReFPO reuses the residual to regularize both signs.

Two controlled comparisons test what this term contributes. Across ten MuJoCo Playground tasks, adding only the regularizer to FPO++ improves mean one-step return by 10.7%. In a one-iteration BallInCup intervention, we hold positive-sample regularization fixed and add the term on negative-advantage samples. CFM error then grows less on held-out positive-advantage actions, consistent with coupling through the shared velocity field. GridWorld illustrates multimodal behavior, and 72-DoF humanoid tracking tests one-step control in a higher-dimensional action space.

2 Related Work

Online optimization of generative policies. DDPO, DPPO, Flow-GRPO, and ReinFlow optimize denoising policies through likelihood-based policy gradients [1, 10, 17, 28]. FPO instead uses a CFM-loss proxy for the old/new action ratio [15]. FPO++ improves this backbone with per-sample ratios and an asymmetric trust-region objective [25]. ReFPO adds a geometric fitting term to the existing residual, complementing changes to the surrogate objective. NCDPO combines backpropagation through diffusion timesteps with behavior-cloning regularization [24]. ReFPO shares the broad goal of preserving fit on policy-generated data, but neither backpropagates through the sampling solver nor introduces a separate imitation buffer or another flow-model evaluation.

Straightening and one-step policies. Rectified Flow learns a velocity field from straight-line interpolations between paired samples. Reflow repeats this procedure using endpoint pairs generated by the previous flow [11]. Consistency and MeanFlow approaches offer other routes to few-step generation [6, 19]. Recent policy methods differ in their optimization setting. TRFP uses off-policy critics, a hybrid ODE/SDE construction, gradient truncation, and flow straightening [29]. FLAME uses Q-reweighted flow matching, an entropy estimator, and a MeanFlow variant for one-step control [9]. FMER combines weighted CFM with mirror descent and entropy constraints [5]. FPMD also obtains one-step policies through off-policy mirror descent [3]. OFP learns one-step visuomotor policies through self-distillation on demonstrations [8]. ReFPO addresses the on-policy old/new action-ratio problem using the same CFM residual as FPO. We compare these settings and their auxiliary machinery in Appendix H; their published scores are not directly comparable across different data, critics, and environments.

3 Reflow-regularized Flow Matching Policy Gradients

3.1 PPO and the FPO proxy

PPO uses the likelihood ratio $\pi_\theta(a \mid o)/\pi_{\text{old}}(a \mid o)$ in a clipped advantage-weighted surrogate [18]. For flow policies, exact action likelihoods generally require costly density evaluation. FPO substitutes

$$r_i = \exp(\ell_i(\theta_{\text{old}}) - \ell_i(\theta)), \qquad L_i^{\text{FPO}} = -\min\{A_i r_i, A_i \,\text{clip}(r_i, 1-\delta, 1+\delta)\}, \tag{1}$$

where i indexes a rollout sample (o_i, a_i, A_i), ℓ_i is its Monte-Carlo CFM loss, and $0 < \delta < 1$ is the clipping parameter. The weighted denoising loss has a likelihood-related variational interpretation [7, 15], but this exponential loss difference is a surrogate, not an exact clean-action likelihood ratio or a ratio of ELBO values.

3.2 Signed local Reflow-style fitting

For M fixed noise/time draws, write

$$\ell_i(\theta) = \frac{\kappa}{Md} \sum_{m=1}^{M} \|v_\theta((1-\tau_{im})\epsilon_{im} + \tau_{im} a_i, \tau_{im}; o_i) - (a_i - \epsilon_{im})\|_2^2 . \tag{2}$$

Here d is the action dimension and $\kappa > 0$ is a fixed proxy-loss scale; $\kappa = 1$ gives the dimension-averaged velocity MSE. We use the noise-to-action convention, in which $\tau \in [0, 1]$ denotes flow time; s denotes optimization time in the parameter updates below.

Update direction. Within one PPO iteration, the action a_i, advantage A_i, cached old loss $\ell_i(\theta_{\text{old}})$, and Monte-Carlo draws are fixed; only θ changes. With $g_i = \nabla_\theta \ell_i$, differentiating $r_i = \exp(\ell_i(\theta_{\text{old}}) - \ell_i(\theta))$ gives $\nabla_\theta r_i = -r_i g_i$. Away from the clipping kinks, the minimum in Equation 1 selects the unclipped term exactly when $A_i > 0$ and $r_i < 1 + \delta$, or $A_i < 0$ and $r_i > 1 - \delta$. We set $c_i = 1$ on this active branch and $c_i = 0$ on the clipped plateau, where the selected term is constant in θ. Hence

$$\nabla_\theta L_i^{\text{FPO}} = -c_i A_i \nabla_\theta r_i = c_i A_i r_i g_i, \qquad \frac{d\ell_i}{ds} = \left\langle g_i, \frac{d\theta}{ds} \right\rangle = -c_i A_i r_i \|g_i\|_2^2, \tag{3}$$

where the second identity is for the isolated-sample gradient flow $d\theta/ds = -\nabla_\theta L_i^{\text{FPO}}$. If $A_i = 0$, the FPO gradient vanishes. On the active branch with $g_i \neq 0$, a positive-advantage sample decreases its CFM loss, fitting the straight target $a_i - \epsilon_{im}$, whereas a negative-advantage sample increases it (anti-CFM ascent); a clipped plateau contributes no FPO geometric update. Equation 3 describes an isolated continuous-time update. Section 3.3 analyzes shared-parameter coupling under gradient flow; neither analysis guarantees the behavior of discrete Adam updates with preconditioning and momentum.

Scope of the Reflow interpretation. Classical Reflow retains a transport pair $(z_0, \Phi_{\text{old}}(z_0))$ and fits its straight chord. FPO instead generates an action and then draws fresh noise/time pairs for its proxy, so the stored noise need not be the latent that generated the action. Equation 3 therefore describes *signed local CFM fitting on the self-generated action marginal with a Reflow-style straight target*, not pathwise or algorithmic equivalence to classical Reflow. When policy improvement succeeds, reward-favored action regions may gain on-policy occupancy and receive repeated aligned fitting; this is an occupancy-level interpretation rather than a convergence theorem (Appendix E).

3.3 Reflow regularization and the shared field

For minibatch size B and fixed $\lambda \geq 0$, ReFPO minimizes

$$L^{\text{ReFPO}} = \frac{1}{B} \sum_i \left(L_i^{\text{FPO}} + \lambda \ell_i \right), \qquad \nabla_\theta (L_i^{\text{FPO}} + \lambda \ell_i) = (c_i A_i r_i + \lambda) g_i, \tag{4}$$

which reuses the current-policy CFM loss already computed for the proxy ratio. For an isolated sample, the same gradient-flow argument gives $d\ell_i/ds = -(c_i A_i r_i + \lambda)\|g_i\|_2^2$. The regularizer therefore accelerates fitting on active positive samples and continues fitting at rate $\lambda \|g_i\|_2^2$ on clipped plateaus, where FPO alone is silent. On an active negative sample, the net coefficient is $\lambda - |A_i| r_i$: if $\lambda < |A_i| r_i$, CFM ascent continues at a reduced rate; at $\lambda = |A_i| r_i$ the sample is locally neutral; only if $\lambda > |A_i| r_i$ is the ascent reversed into descent. The advantage and policy-gradient term are unchanged, but the net update balances reward-directed rejection against CFM fitting; sufficiently strong regularization can reverse the isolated negative-sample update.

Shared-field coupling. All samples share θ, so these single-sample statements do not transfer directly to a minibatch. Under the batch gradient flow $d\theta/ds = -\nabla_\theta L^{\text{ReFPO}}$,

$$\frac{d\ell_i}{ds} = -\frac{1}{B} \sum_j (c_j A_j r_j + \lambda) \langle g_i, g_j \rangle. \tag{5}$$

The self term $j = i$ recovers the isolated rate up to the factor $1/B$, but cross terms can have either sign. When $\langle g_i, g_j \rangle > 0$, the anti-CFM push of an active negative sample j raises the error of sample i, even

if $A_i > 0$, and adding λ shrinks that push. Averaging Equation 5 over i, with $G_C = \nabla_\theta B^{-1} \sum_i \ell_i$ and $G_F = \nabla_\theta B^{-1} \sum_i L_i^{\mathrm{FPO}}$, gives the batch identity

$$\frac{d}{ds}\frac{1}{B}\sum_i \ell_i = -\langle G_C, G_F\rangle - \lambda\|G_C\|_2^2. \tag{6}$$

The regularizer contributes the nonpositive restoring term $-\lambda\|G_C\|_2^2$, while the cross term can have either sign, so the total can still increase; with signed weights, clipping, and changing rollouts there is no general monotonicity guarantee. Whether regularizing negative samples protects the positive part of the shared field is therefore an empirical question, which Section 4.3 tests directly.

One-step inference. For $\dot{x}(\tau) = v_\theta(x(\tau), \tau; o)$, hold o and θ fixed and apply the chain rule:

$$\frac{d}{d\tau}v_\theta = \partial_\tau v_\theta + J_x v_\theta \, v_\theta, \qquad \|\mathrm{LTE}\| \le \frac{h^2}{2}\left(\|\partial_\tau v_\theta\| + \|J_x v_\theta\|_{\mathrm{op}}\|v_\theta\|\right) + O(h^3). \tag{7}$$

Here LTE is Euler's one-step state error starting from the exact state, and $\|\cdot\|_{\mathrm{op}}$ is the induced Euclidean operator norm. The expansion assumes local smoothness and a bounded third derivative of the solution (Appendix E). Straight-target fitting motivates coarse-step inference, but small conditional residuals alone do not bound these derivatives along every generated path, so we evaluate one-step control empirically.

Proxy-ratio sensitivity. Since $\log r_i = \ell_i(\theta_{\mathrm{old}}) - \ell_i(\theta)$, Equation 5 also governs ratio motion: an isolated sample has $d\log r_i/ds = (c_i A_i r_i + \lambda)\|g_i\|_2^2$, so the regularizer increases its immediate upward ratio movement, while minibatch cross terms can move r_i in either direction.

To make the local sensitivity explicit, stack the M velocity residuals in Equation 2 and multiply by $\sqrt{\kappa/(Md)}$ to obtain e_i. With parameter Jacobian $J_i = \partial e_i/\partial\theta$,

$$\ell_i = \|e_i\|_2^2, \qquad g_i = \nabla_\theta \ell_i = 2J_i^\top e_i.$$

For a small parameter update $\delta\theta$, hold the old loss and CFM draws fixed and assume that ℓ_i has a locally bounded Hessian. Taylor expansion, followed by Cauchy–Schwarz and the operator-norm inequality, gives

$$\begin{aligned} \Delta\log r_i &= -\langle g_i, \delta\theta\rangle + O(\|\delta\theta\|_2^2), \\ |\Delta\log r_i| &\le 2\|J_i^\top e_i\|_2\|\delta\theta\|_2 + O(\|\delta\theta\|_2^2) \\ &\le 2\|J_i\|_{\mathrm{op}}\|e_i\|_2\|\delta\theta\|_2 + O(\|\delta\theta\|_2^2). \end{aligned} \tag{8}$$

The quantities on the right are evaluated before the update. Smaller residuals reduce the leading sensitivity bound under comparable Jacobians and update sizes; this does not by itself guarantee smaller realized ratio changes.

Upper and lower ratio tails. Write $\ell_{i,\mathrm{old}} = \ell_i(\theta_{\mathrm{old}})$ and $\ell_{i,\mathrm{new}} = \ell_i(\theta)$. Nonnegativity of the CFM loss yields

$$\begin{aligned} r_i \le e^{\ell_{i,\mathrm{old}}}, \qquad r_i > e^3 &\implies \ell_{i,\mathrm{old}} > 3, \\ r_i < e^{-3} &\iff \ell_{i,\mathrm{new}} > \ell_{i,\mathrm{old}} + 3. \end{aligned} \tag{9}$$

These relations distinguish two effects. Across outer iterations, smaller behavior-policy residuals lower the upper bound $e^{\ell_{i,\mathrm{old}}}$. Within an inner update, the old loss is fixed, so decreasing the current loss instead increases r_i. Lower-tail excursions correspond to current-loss growth, which the unweighted CFM term opposes through its restoring contribution in Equation 6; cross-sample interference can still raise individual losses. We therefore report the *two-sided log-proxy-ratio excursion rate*, $\Pr(|\log r| > 3)$, and do not infer separate upper-tail control or a unique causal mechanism for training stability from this aggregate statistic.

4 Experiments

We evaluate ReFPO across three domains: a multi-goal GridWorld task to visualize multimodal control [2, 22], MuJoCo Playground [21, 27] for standard continuous control, and high-dimensional Humanoid Control for motion tracking with varying goal information. The vanilla-FPO benchmark

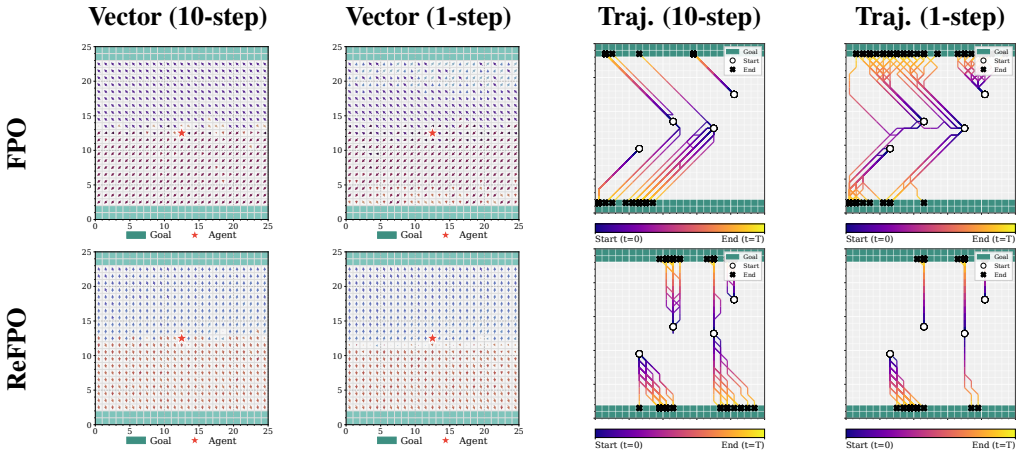


Figure 2: **Visualizations of the multimodal GridWorld task.** Top: FPO; bottom: ReFPO. Left: sampled policy action fields over environment states. Right: environment-state rollouts from fixed initial states. Neither depicts the internal noise-to-action ODE path used to generate each action. Reflow straightness concerns that internal path and does not require a vertical state rollout. Enlarged action fields appear in Appendix J.

comparisons and Humanoid results use five random seeds; the FPO++ comparison uses ten seeds per task. MuJoCo reports averages of taskwise seed means and SDs, whereas Humanoid reports mean and SD over full-set seed evaluations. The five-paired-seed sign-conditioned intervention instead reports standard errors (SE); the targeted late-drop trajectories are individual runs. These aggregation units are specified with each result.

Coefficient convention. Write $\ell_i = \kappa C_i$, where C_i is the raw dimension-averaged CFM error. If an implementation adds $\lambda_{\text{raw}} C_i$, the coefficient in Equation 4 is $\lambda = \lambda_{\text{raw}}/\kappa$. Playground uses $\kappa = 1$; the Humanoid velocity proxy uses $\kappa = 200$, so its raw coefficient 0.1 corresponds to $\lambda = 0.0005$. All λ values in our tables and hyperparameter lists are the reported raw coefficients (Appendix E).

4.1 GridWorld

We evaluate ReFPO in a 25×25 GridWorld with bifurcated sparse rewards to test whether it preserves multimodal action distributions while improving discretization robustness. We compare against vanilla FPO [15].

Multimodality and Path Geometry. Figure 2 visualizes sampled action fields and environment-state rollouts of FPO and ReFPO under $N = 10$ Euler steps. ReFPO preserves the multimodal structure of the policy: trajectories still bifurcate near the central saddle point and reach both goal regions. ReFPO produces more coherent action directions while retaining both goal-reaching modes in this visualization. This is qualitative task behavior; internal flow geometry is assessed separately using the CFM diagnostic.

One-Step Inference. Figure 2 also compares one-step ($N = 1$) generation. Vanilla FPO can already complete this simple task with a single Euler step, showing that one-step capability is already present in this baseline. However, its one-step action field still exhibits local fluctuations and directional drift, which can make coarse discretization less reliable. ReFPO reduces these artifacts and produces more coherent one-step samples, indicating that explicit Reflow regularization improves discretization robustness without collapsing the multimodal structure of the policy.

4.2 MuJoCo Playground

We evaluate ReFPO on 10 continuous-control tasks adapted from the DeepMind Control Suite [20, 23] within MuJoCo Playground, testing standard online control performance and low-latency one-step inference.

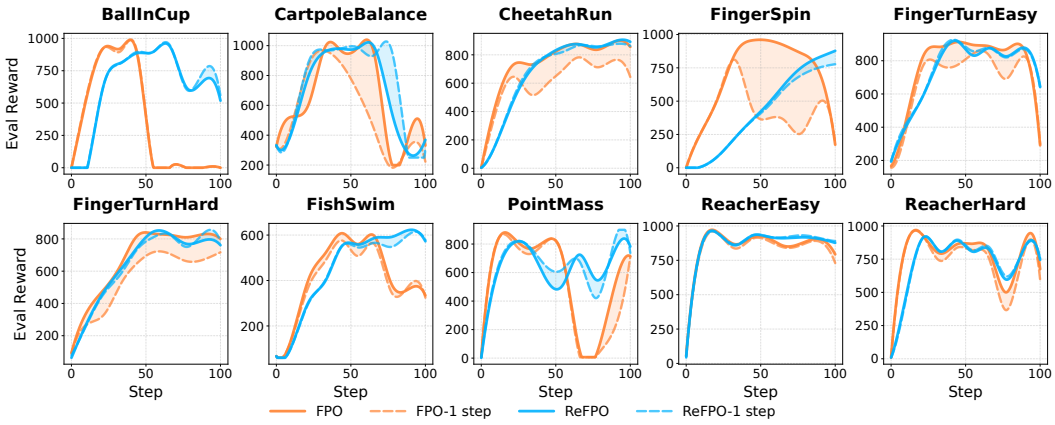


Figure 3: **MuJoCo Playground evaluation.** FPO and ReFPO rewards on 10 DM Control Suite tasks over 100M steps. Shaded regions show the 10-step/1-step gap; narrower gaps indicate better one-step consistency.

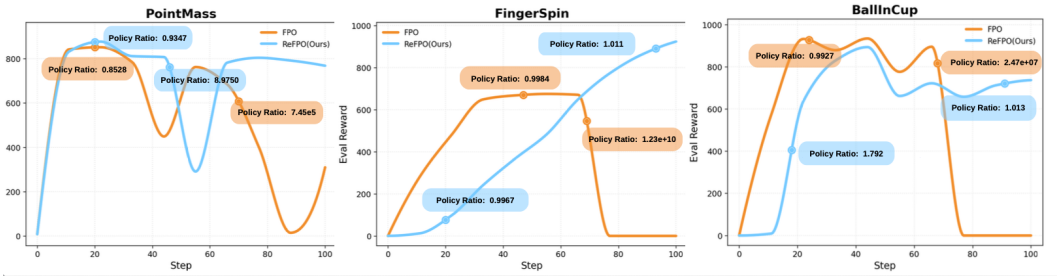


Figure 4: **Policy-ratio diagnostics.** Selected PointMass, FingerSpin, and BallInCup traces illustrate the association between proxy-ratio fluctuations and reward degradation. The aggregate two-sided excursion statistic is reported separately in Table 1; these traces do not identify a unique causal mechanism.

Experimental Setup and Metrics. Following FPO [15], we use Adam, batch size 1024, and 16 updates per batch. The train-from-scratch comparison uses 100M environment steps; flow policies are evaluated with $N = 1$ and $N = 10$ inference steps. Since online RL for flow-matching policies is recent and directly comparable baselines remain limited, we use FPO as the closest likelihood-free flow-policy baseline and include Gaussian PPO as a strong non-generative control baseline. We further compare with ReinFlow and Flow-GRPO, two flow-matching post-training baselines, using the same tasks and tuning budget. Table 1 reports their 100M results and, for the best configurations found during tuning, their results after a further 100M training steps (200M total), with details in Section B.2.

Supervision modes. FPO and ReFPO share all training hyperparameters except λ and the CFM supervision mode. FPO uses noise supervision, the stronger of the two FPO variants we evaluated (Table B.7), whereas ReFPO uses velocity supervision, the residual on which its regularizer is defined. Table 1 is therefore not a fully regularizer-isolated comparison. The noise-supervised proxy also weights velocity residuals by $(1-t)^2$ (Appendix E), so equal excursion thresholds across these modes do not represent identical velocity-error changes. The FPO++ comparison below and the sign-conditioned intervention in Section 4.3 provide controlled evidence for the regularizer.

Beyond cumulative episode return in environment units, we report the CFM velocity MSE of Equation 2 with $\kappa = 1$ in normalized action space, averaged over evaluation samples and flow times. This "Straightness MSE" measures agreement with a constant conditional velocity, not deviation of a generated trajectory from its endpoint chord. The excursion rate is

$$\Pr(|\log r| > 3) = \Pr(r > e^3 \text{ or } r < e^{-3}); \qquad 0.00189 = 0.189\%.$$

Table 1: **MuJoCo Playground results on ten tasks.** Each method–task uses five seeds and 128 episodes per seed at each inference setting. Means average the ten taskwise seed means; $\pm$ averages their cross-seed SDs. The last column is the dimensionless fraction $\Pr(|\log r| > 3)$. ReFPO* uses $\lambda = 0.04$. $N = 1$ is a forced one-step diagnostic for the flow baselines. Gaussian PPO samples actions directly; its return is shown in the first reward column for reference.

Method	10-step Reward	1-step Reward	CFM MSE	Excursion
Training stage: 0–100M steps				
PPO	622 ± 192	–	–	–
Flow-GRPO	475 ± 194	$(1.44 \pm 2.18) \times 10^{-4}$	–	–
ReinFlow	416 ± 195	339 ± 194	–	–
FPO	641 ± 140	565 ± 160	0.0475	0.00365
ReFPO*	$\mathbf{686 \pm 139}$	$\mathbf{690 \pm 139}$	**0.0116**	**0.00189**
ReFPO, $\lambda = 0.02$	644 ± 132	556 ± 160	0.0187	0.00377
ReFPO, $\lambda = 0.08$	646 ± 141	652 ± 151	0.0161	0.00569
ReFPO, $\lambda = 0.12$	609 ± 147	601 ± 155	0.0193	0.00330
ReFPO, Trajectory-advantage-weighted	671 ± 161	640 ± 166	–	–
Total budget: 200M steps (continued from 100M)				
Flow-GRPO	507 ± 235	$(1.50 \pm 2.26) \times 10^{-4}$	–	–
ReinFlow	523 ± 188	357 ± 178	–	–

It counts both tails, including small ratios. Each method and inference setting uses $10 \times 5 \times 128 = 6{,}400$ evaluation episodes with matched initializations. The $N = 10$ and $N = 1$ evaluations use the same run/checkpoint. FPO/ReFPO use ten sampling steps during training and eight CFM draws per action. Baseline-specific configurations appear in Appendix B.2.

Returns and comparative stability. Table 1 shows higher average 10-step and one-step returns for ReFPO than for FPO, with reductions of 75.6% in Straightness MSE and 48.2% in two-sided excursions. Figure 3 also shows remaining late-stage drops in some tasks. These are comparative improvements, not monotonic-training guarantees. One-step returns can match or slightly exceed ten-step returns because control return and numerical accuracy for a learned ODE are distinct quantities; the results do not imply that one-step integration is generally more accurate.

Flow-GRPO's near-zero one-step values in Table 1 are retained from the evaluation outputs. Its policy is optimized over ten denoising transitions; forcing one step changes the trained inference regime. Failed control can still receive tiny positive shaped rewards. The 200M rows are total-budget continued-training results, not an additional 200M steps.

Complementarity with FPO++. We add the same unweighted CFM regularizer to FPO++ [25], obtaining ReFPO++ with $\lambda = 0.02$. Both methods use the same architectures, velocity-field parameterizations, optimizers, training budgets, checkpoint-selection rule, and evaluators; the only algorithmic difference is the additional regularizer. Each task uses ten seeds and 128 evaluation episodes per seed and inference setting. The paired one-step and ten-step evaluations use the same run and checkpoint.

Table 2: **Reflow regularization on FPO++.** Means average the ten taskwise ten-seed means; $\pm$ averages the ten taskwise cross-seed SDs. Diagnostics are averaged over the same task–seed runs. The excursion column is the dimensionless fraction $\Pr(|\log r| > 3)$.

Method	10-step Reward	1-step Reward	CFM MSE	Excursion
FPO++	665.85 ± 125.52	639.66 ± 122.05	0.002173	0.000418587
ReFPO++	$\mathbf{710.23 \pm 135.89}$	$\mathbf{708.42 \pm 131.67}$	**0.000534**	**0.000291645**

Table 2 shows that the regularizer improves ten-step reward by 6.7% and one-step reward by 10.7%, while reducing CFM MSE by 75.4% and two-sided proxy-ratio excursions by 30.3%. These results indicate that the same regularizer complements the more stable FPO++ policy-gradient objective.

Regularization scope. Among the coefficients reported in Table 1, $\lambda = 0.04$ gives the strongest aggregate reward–geometry trade-off. The effects are not monotonic: $\lambda = 0.02$ and 0.08 have higher excursion rates than FPO, and $\lambda = 0.02$ also has lower one-step return. Thus the stability improvement is coefficient-dependent. The trajectory-advantage-weighted variant obtains lower returns than unweighted ReFPO, but does not isolate the negative branch; Section 4.3 isolates that branch directly. The BallInCup and CartpoleBalance traces discussed in Appendix G are individual runs with different seeds, not seed-averaged curves or a controlled estimate of the coefficient effect.

4.3 Sign-conditioned intervention

To isolate the negative-advantage component of the regularizer, define the sign-masked terms

$$L_R^{\pm} = \frac{1}{B} \sum_i \mathbf{1}(A_i \gtrless 0)\, \ell_i,$$

both with the full-batch denominator; these masks omit exactly zero advantages. With L_{PG} the common policy loss, we compare positive-only regularization, $L_{\mathrm{PG}} + \lambda L_R^+$, with regularization on both signs, $L_{\mathrm{PG}} + \lambda(L_R^+ + L_R^-)$. Positive-sample weights are identical, so the objectives differ only by the negative-sample regularizer. For each of five paired BallInCup seeds, both arms start from the same checkpoint, rollout batch, CFM draws, Adam state, and minibatch order, and perform the normal 16 inner epochs at $\lambda = 0.04$. We also evaluate a disjoint held-out next rollout from the unchanged pre-update policy, with 983,040 transitions per seed and no optimization on those transitions.

Table 3: **Sign-conditioned BallInCup intervention.** Mean $\pm$ SE over five paired seeds, in units of 10^{-3}; these SEs differ from the SDs in the benchmark tables. Here $\Delta\ell = \ell_{\mathrm{current}} - \ell_{\mathrm{old}}$, and superscripts denote the advantage sign. A negative paired difference means less CFM-error growth after adding only L_R^-. Each improvement direction holds in all five paired seeds.

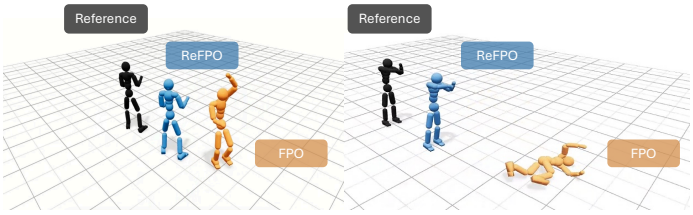
Measurement	Mean $\pm$ SE
Update-batch $\Delta\ell^-$, positive-only	0.995 ± 0.309
Update-batch $\Delta\ell^-$, both signs	0.703 ± 0.323
Held-out $\Delta\ell^-$, both signs minus positive-only	-0.320 ± 0.094
Held-out $\Delta\ell^+$, both signs minus positive-only	-0.236 ± 0.044

Table 3 shows that adding L_R^- reduces negative-advantage CFM growth on the update batch by 29.3%, while the average growth remains positive. On the held-out rollout, the added term lowers error growth on negative-advantage actions and also on positive-advantage actions, although it is evaluated only on negative-advantage training samples. This supports protection of the positive part of the shared field through the shared-parameter coupling described in Section 3.3. The study measures a local, one-iteration effect, not long-horizon return. Five same-sign paired effects alone do not establish $p < 0.05$ under a two-sided sign test. Appendix F gives details and a seed-level example.

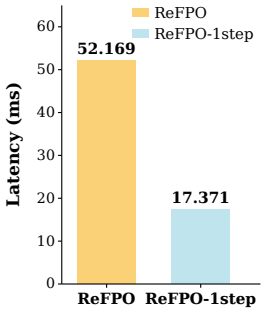
4.4 Humanoid Control

Physics-aware humanoid control provides a high-dimensional benchmark for evaluating the scalability of flow-based policies. We consider a MoCap tracking task in which a simulated SMPL-based agent must reproduce diverse AMASS reference motions [13] while maintaining physical balance and coordinated full-body control. This setting stresses both the expressivity of the action distribution and the discretization robustness required for low-latency inference. As shown in Figure 5a, ReFPO tracks the reference motion more stably than FPO, with additional examples provided in the supplementary materials.

Implementation Details and Metrics. The agent has 24 actuated joints (72 DoF) and is trained in Isaac Gym [14] using Puffer-PHC [12]. Policies receive proprioception and goal-conditioned



(a) Performance on Humanoid Control.



(b) Inference latency comparison.

Figure 5: **Performance and efficiency analysis.** (a) Comparison of FPO and ReFPO on Humanoid Control. (b) Action generation latency, where ReFPO-1step reduces action-generation time.

targets under three conditioning regimes: full-joint, root+hands, and root-only; in the latter two sparse-conditioning settings, multimodal action modeling becomes more important. All policies are trained for 100M environment steps with the imitation reward from [16].

Table 4: **Quantitative evaluation on Humanoid Control tasks.** Mean $\pm$ SD over five complete 100M-step runs per method and conditioning regime, with all 11,313 AMASS motions evaluated per seed and no seed selection. Success-rate uncertainties are percentage points. Success is completion of a reference motion without early termination. Metrics are Success Rate (↑), Alive Duration (↑), and MPJPE in millimetres (↓).

Methods	Goal conditioning	Success rate (↑)	Alive duration (↑)	MPJPE (↓)
Gaussian PPO	All joints	$98.7\% \pm 1.2\%$	200.46 ± 5.21	31.62 ± 2.17
FPO	All joints	$96.2\% \pm 0.8\%$	197.64 ± 3.94	42.65 ± 1.73
FPO-1step	All joints	$96.9\% \pm 0.6\%$	198.25 ± 4.18	41.20 ± 1.62
ReFPO	All joints	$96.6\% \pm 0.9\%$	198.03 ± 3.71	41.00 ± 1.91
ReFPO-1step	All joints	$97.3\% \pm 0.7\%$	198.47 ± 4.33	39.49 ± 1.55
Gaussian PPO	Root + Hands	$46.5\% \pm 2.9\%$	142.50 ± 4.72	97.65 ± 2.08
FPO	Root + Hands	$71.0\% \pm 2.0\%$	171.68 ± 4.08	63.88 ± 1.76
FPO-1step	Root + Hands	$70.8\% \pm 1.7\%$	171.37 ± 3.85	64.96 ± 1.94
ReFPO	Root + Hands	$72.7\% \pm 2.3\%$	174.98 ± 4.41	62.68 ± 1.68
ReFPO-1step	Root + Hands	$74.4\% \pm 1.8\%$	174.83 ± 3.97	60.60 ± 2.03
Gaussian PPO	Root	$29.8\% \pm 3.2\%$	114.06 ± 5.16	123.70 ± 2.31
FPO	Root	$55.0\% \pm 2.6\%$	147.94 ± 4.36	73.55 ± 1.88
FPO-1step	Root	$55.3\% \pm 2.3\%$	147.90 ± 3.92	72.65 ± 1.71
ReFPO	Root	$58.2\% \pm 2.8\%$	156.37 ± 4.57	70.97 ± 2.06
ReFPO-1step	Root	$58.7\% \pm 2.2\%$	156.45 ± 4.14	70.89 ± 1.79

Results and Qualitative Analysis. Table 4 shows that ReFPO maintains strong one-step performance in high-dimensional humanoid control. Across the flow-policy variants, the $N = 1$ results are consistently comparable to, and sometimes better than, their $N = 10$ counterparts, while Figure 5 shows the expected latency reduction from one-step action generation. This supports the practical motivation of ReFPO: high-quality actions can be generated with substantially lower inference cost.

The results also clarify where expressive flow policies are most useful. Under full-joint conditioning, Gaussian PPO remains the strongest method, indicating that a simple Gaussian policy is highly competitive when the target information is dense. As conditioning becomes sparser, however, the advantage shifts toward flow-based policies: FPO and ReFPO substantially outperform Gaussian PPO in the root+hands and root-only settings. ReFPO further improves over FPO in these sparse regimes, with its one-step variant achieving the best success rate and MPJPE. These results suggest that ReFPO is especially beneficial when incomplete goal information makes multimodal action modeling and robust one-step inference important.

High-dimensional diagnostics. For the 72-DoF Root-only setting, ReFPO reduces the two-sided excursion rate from 0.2429818% to 0.0388826% (84.0%) and the NFE=100 Straightness MSE from 1.1050×10^{-3} to 1.0499×10^{-3} (4.99%). Each rate uses all finite training samples through the common 100M-step cutoff within a seed and is then averaged over five seeds. The implementation's raw-CFM threshold 0.015 corresponds to $|\log r| > 3$ after scaling by $1/(2\sigma^2) = 200$, with $\sigma = 0.05$. These results support comparative stabilization in this setting; the ratio statistic does not separately measure its upper tail.

5 Discussion and Limitations

ReFPO specializes self-imitation to FPO's CFM proxy through an advantage-independent geometric regularizer. Its contribution combines the signed local analysis, the coefficient shift in Equation 4, and evidence linking shared-field fitting with improved one-step control and proxy stability. The evaluated gains build on FPO's existing one-step capability.

Our analysis holds for frozen inner-loop samples and does not establish equivalence to classical transport-pair Reflow, global convergence, or universal upper-tail suppression. Straight-target CFM error is a diagnostic, not a bound on all generated ODE paths. Some tasks still exhibit late-stage degradation; targeted coefficient examples do not establish a seed-controlled tuning benefit. The five-paired-seed sign-conditioned study measures local field protection, not a universal long-horizon return improvement. Full-joint humanoid tracking remains a setting where Gaussian PPO is strongest. Generalization to other policy families, adaptive regularization, and robust task-specific tuning remain open.

Acknowledgments and Disclosure of Funding

This work was supported in part by the NSFC (Grant Nos. 62573371, 62293482, 61931024, 12326610, and 62501514), the Guangdong S&T Programme (Grant No. 2024B0101030002), the Basic Research Project of Hetao Shenzhen-HK ST Cooperation Zone (No. HZQBKCZYZ-2021067), and the Shenzhen Outstanding Talents Training Fund (No. 202002). This work was also supported by the National Key Research and Development Program of China (Grant Nos. 2025YFF0515300 and 2025YFF0515304), the Guangdong Province Radio Science Data Center (Grant No. 2025B1212070001), the Shenzhen General Program (No. JCYJ20220530143600001), the Guangdong Research Projects (Nos. 2017ZT07X152 and 2019CX01X104), the Guangdong Provincial Key Laboratory of Future Networks of Intelligence (Grant No. 2022B1212010001), GRDP2026-059, the Shenzhen Key Laboratory of Big Data and Artificial Intelligence (Grant No. SYSPG20241211173853027), the Open Project Program of Key Laboratory of Tibetan Information Processing, Ministry of Education (Grant No. QHSFCS-2606), the Shenzhen-Hong Kong Joint Funding (No. SGDX20211123112401002), the Tencent & Huawei Open Fund (Nos. 2024E0009 and 202301030019), and the Shenzhen Municipal-Level Talent Program.

The authors declare no additional competing interests.

References

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

[2] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym. *arXiv preprint arXiv:1606.01540*, 2016.

[3] Tianyi Chen, Haitong Ma, Na Li, Kai Wang, and Bo Dai. One-step flow policy mirror descent, 2025. URL https://arxiv.org/abs/2507.23675.

[4] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

[5] Ting Gao, Stavros Orfanoudakis, Nan Lin, Winnie Daamen, Serge Hoogendoorn, and Elvin Isufi. Flow matching policy optimization with mirror descent and entropy constraints, 2026. URL https://arxiv.org/abs/2603.17685.

[6] Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

[7] Diederik P Kingma and Ruiqi Gao. Understanding diffusion objectives as the ELBO with simple data augmentation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=NnMEadcdyD.

[8] Shaolong Li, Lichao Sun, and Yongchao Chen. One-step flow policy: Self-distillation for fast visuomotor policies, 2026. URL https://arxiv.org/abs/2603.12480.

[9] Zeqiao Li, Yijing Wang, Haoyu Wang, Zheng Li, and Zhiqiang Zuo. Boosting maximum entropy reinforcement learning via one-step flow matching, 2026. URL https://arxiv.org/abs/2602.01606.

[10] Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-GRPO: Training flow matching models via online RL. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

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

[12] Zhengyi Luo, Jinkun Cao, Kris Kitani, Weipeng Xu, et al. Perpetual humanoid control for real-time simulated avatars. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10895–10904, 2023.

[13] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. AMASS: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5442–5451, 2019.

[14] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.

[15] David McAllister, Songwei Ge, Brent Yi, Chung Min Kim, Ethan Weber, Hongsuk Choi, Haiwen Feng, and Angjoo Kanazawa. Flow matching policy gradients. In *The Fourteenth International Conference on Learning Representations*, 2026.

[16] Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel Van de Panne. DeepMimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions On Graphics (TOG)*, 37(4):1–14, 2018.

[17] 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*, 2025. URL https://openreview.net/forum?id=mEpqHvbD2h.

[18] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[19] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 32211–32252, 2023.

[20] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

[21] Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

[22] Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, et al. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024.

[23] Saran Tunyasuvunakool, Alistair Muldal, Yotam Doron, Siqi Liu, Steven Bohez, Josh Merel, Tom Erez, Timothy Lillicrap, Nicolas Heess, and Yuval Tassa. dm_control: Software and tasks for continuous control. *Software Impacts*, 6:100022, 2020.

[24] Ningyuan Yang, Jiaxuan Gao, Feng Gao, Yi Wu, and Chao Yu. Fine-tuning diffusion policies with backpropagation through diffusion timesteps, 2025. URL https://arxiv.org/abs/2505.10482.

[25] Brent Yi, Hongsuk Choi, Himanshu Gaurav Singh, Xiaoyu Huang, Takara E. Truong, Carmelo Sferrazza, Yi Ma, Rocky Duan, Pieter Abbeel, Guanya Shi, Karen Liu, and Angjoo Kanazawa. Flow policy gradients for robot control, 2026. URL https://arxiv.org/abs/2602.02481.

[26] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6613–6623, 2024.

[27] Kevin Zakka, Baruch Tabanpour, Qiayuan Liao, Mustafa Haiderbhai, Samuel Holt, Jing Yuan Luo, Arthur Allshire, Erik Frey, Koushil Sreenath, Lueder A Kahrs, et al. MuJoCo playground. *arXiv preprint arXiv:2502.08844*, 2025.

[28] Tonghe Zhang, Chao Yu, Sichang Su, and Yu Wang. Reinflow: Fine-tuning flow matching policy with online reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[29] Xubin Zhou, Yipeng Yang, and Zhan Li. Truncated rectified flow policy for reinforcement learning with one-step sampling, 2026. URL https://arxiv.org/abs/2604.09159.

A Code and Supplementary Videos

The supplementary materials contain the ReFPO training, configuration, and paired $N = 10/N = 1$ evaluation code. Algorithm D.1 describes the update, and Appendix I gives the implementation-level loss addition.

Additionally, we provide supplementary videos demonstrating the performance of ReFPO and FPO across MuJoCo Playground and Humanoid Control tasks. These videos showcase the high-fidelity trajectories achieved during one-step inference ($N = 1$) and provide direct comparisons with the baseline methods.

Third-party assets and access. The upstream simulator and benchmark code includes MuJoCo Playground (Apache License 2.0; `https://github.com/google-deepmind/mujoco_playground`) and PHC (BSD 3-Clause Clear License; `https://github.com/ZhengyiLuo/PHC`). Isaac Gym is subject to the NVIDIA Isaac Gym License Agreement (`https://developer.nvidia.com/isaac-gym/download`). AMASS and SMPL are available under their respective non-commercial scientific research licenses (`https://amass.is.tue.mpg.de/license.html`; `https://smpl.is.tue.mpg.de/modellicense.html`). These data and body-model assets must be obtained from their original providers under the applicable terms; they are not included in our code release. Third-party components retain their own licenses.

B Training Settings

B.1 GridWorld

All GridWorld experiments were conducted using an **NVIDIA GeForce RTX 4090 GPU**. To ensure a rigorous and fair comparison, we maintained identical hyperparameters for both the vanilla FPO baseline and our proposed ReFPO method with 5 seeds, the only exception being the Reflow regularization coefficient λ utilized in ReFPO. Each run required approximately 0.5 GPU hours, resulting in roughly 5 RTX 4090 GPU hours in total for the GridWorld experiments. A comprehensive summary of these hyperparameters is provided in Table B.5.

Table B.5: Hyperparameters for GridWorld Experiments

HYPERPARAMETER	VALUE
OPTIMIZER	ADAM
LEARNING RATE	3×10^{-4}
TOTAL TIMESTEPS	260,000
TIMESTEPS PER BATCH	2048
MINIBATCH SIZE	341 (APPROX. $2048/6$)
UPDATES PER ITERATION	10
MAX EPISODE LENGTH	200
DISCOUNT FACTOR (γ)	0.99
GAE PARAMETER (λ_{GAE})	0.98
PPO CLIP RANGE (δ)	0.2
MAX GRAD NORM	0.5
MC SAMPLES FOR CFM (N_{mc})	50
REFLOW COEFFICIENT	0.1 (0.0 FOR FPO)
ODE SOLVER	EULER
INFERENCE STEPS (N)	1, 10, 20

B.2 MuJoCo Playground

For the MuJoCo Playground benchmarks, we evaluated our methods across 10 continuous control tasks using an **NVIDIA GeForce RTX 4090 GPU**. The base comparison uses 100 million environment steps; the continued-training baseline rows use 200 million total steps. For PPO, we utilized a standard Gaussian policy as a baseline, training it for the same 100 million environment steps with 5 seeds to align the experimental configuration across all methods. Each task-seed-configuration run

Table B.6: **Hyperparameters for PPO, FPO/ReFPO, Flow-GRPO, and ReinFlow.**

(a) PPO

Hyperparameter	**Value**
Discount factor (γ)	0.995
GAE λ	0.95
Value loss coefficient	0.25
Entropy coefficient	0.01
Reward scaling	10.0
Normalize advantage	True
Normalize observations	True
Action repeat	1
Unroll length	30
Batch size	1024
Number of minibatches	32
Number of updates per batch	16
Number of timesteps	100M
Policy network	MLP (4, 32)
Value network	MLP (5, 256)
Optimizer	Adam

(b) FPO / ReFPO

Hyperparameter	**Value**
Discount factor (γ)	0.995
GAE λ	0.95
Value loss coefficient	0.25
Clipping δ	0.05
Reward scaling	10.0
Reflow coefficient	0.04 (ReFPO)
Normalize advantage	True
Normalize observations	True
Flow steps (T)	10
Samples per action	8
Supervision mode	velocity/noise
Time embed dim	8
Unroll length	30
Batch size	1024
Number of minibatches	32
Number of updates per batch	16
Number of timesteps	100M
Policy network	MLP (4, 32)
Value network	MLP (5, 256)
Optimizer	Adam

(c) Flow-GRPO

Hyperparameter	**Value**
Discount factor (γ)	0.995
Reward scaling	10.0
Normalize observations	True
Flow steps (T)	10
Output mode	u
Time embed dim	8
Loss mode	denoising MDP
Group size	8
Advantage clipping max	8.0
Clipping δ	0.1
KL coefficient β	5×10^{-3}
Noise level	1.0
Feather std	0.05
Policy output scale	0.5
Unroll length	30
Number of environments	2048
Batch size	1024
Number of minibatches	32
Number of updates per batch	8
Number of timesteps	200M
Learning rate	1×10^{-4}
Max gradient norm	1.0
Policy network	MLP (4, 32)
Value network	N/A
Optimizer	Adam

(d) ReinFlow

Hyperparameter	**Value**
Discount factor (γ)	0.99
GAE λ	0.95
Learning rate	3×10^{-5}
Optimizer	Adam
Batch size	1024
Number of environments	2048
Unroll length	30
Number of minibatches	32
Number of updates per batch	8
Number of timesteps	200M
Reward scaling	3.0
Normalize observations	True
Normalize advantage	True
Value loss coefficient	0.6
Entropy coefficient	0.05
PPO clipping δ	0.10
Horizon steps	4
Inference denoising steps	10
Fine-tuning denoising steps	2
Timestep embedding dim	16
Action range	$[-1, 1]$
Noise scheduler	Linear
Sampling denoising std min	0.10
Log-prob denoising std range	$[0.08, 0.16]$

required approximately 0.08 GPU hours on average. The original MuJoCo experiment suite was estimated at approximately 44 RTX 4090 GPU hours. This estimate is not a project-wide total: it excludes the additional FPO++ comparison, sign-conditioned interventions, and targeted late-stage runs, for which a consolidated compute total is unavailable.

FPO and ReFPO use the same training hyperparameters except for the regularization coefficient and the CFM supervision mode. The main FPO baseline predicts velocity but supervises it through the reconstructed noise (`u_but_supervise_as_eps`); ReFPO directly supervises velocity (`u`). The output-mode ablation below records the performance of both FPO variants. These supervision choices can affect optimization and should be considered when interpreting the main comparison. For ReFPO, we applied a constant regularization coefficient of $\lambda = 0.04$, which was empirically found to provide the best balance between path straightness and policy expressivity in these continuous-control tasks. The detailed hyperparameter configurations are summarized in Table B.6, where we also include the GRPO and ReinFlow settings.

Table B.7 compares FPO's two supervision modes. Its noise-supervised variant, used in the main comparison, has higher returns and slightly better diagnostics than direct velocity supervision in this evaluation. ReFPO uses direct velocity supervision because its regularizer is defined on that residual. The main comparison therefore uses the stronger evaluated FPO variant, but does not isolate the regularizer from every supervision-mode effect. The sign-conditioned intervention in Section 4.3 holds the model and supervision fixed while isolating the negative-advantage term.

Table B.7: Output parameterization ablation for FPO on MuJoCo Playground.

Method	**10-step Reward**	**1-step Reward**	**CFM MSE**	**Excursion**
FPO	641 ± 140	565 ± 160	0.0475	0.00365
FPO-velocity	602 ± 168	533 ± 172	0.0515	0.00389

B.3 Humanoid Control

All humanoid control experiments were conducted using an **NVIDIA A100 GPU**, with each individual training run requiring approximately **14 hours** to complete. To ensure a rigorous and fair comparison, we maintained identical hyperparameters for both the vanilla FPO baseline and our proposed ReFPO method, with the only exception being the Reflow regularization coefficient λ associated with the Reflow loss, which was utilized exclusively in ReFPO. A comprehensive summary of these hyperparameters is provided in Table B.8.

Table B.8: Hyperparameters for humanoid control.

Hyperparameter	**Value**	**Hyperparameter**	**Value**
Policy Settings			
Hidden size	512	Solver step size	0.1
Action perturbation std	0.05	Target KL divergence	None
Number of environments	4096	Normalize advantage	True
Reflow coefficient λ_{raw}	0.1 for ReFPO; 0.0 for FPO		
Training Settings			
Batch size	131072	Minibatch size	32768
Learning rate	0.0001	LR annealing	False
LR decay rate	1.5e-4	LR decay floor	0.2
Update epochs	4	L2 regularization coef.	0.0
GAE lambda	0.2	Discount factor (γ)	0.98
Clipping coefficient	0.01	Value function coefficient	1.2
Clip value loss	True	Value loss clip coefficient	0.2
Max gradient norm	10.0	Entropy coefficient	0.0
Discriminator coefficient	5.0	Bound coefficient	10.0

C Additional Stability Diagnostics

We include Jacobian and policy-ratio diagnostics as additional stability measures for the MuJoCo Playground experiments. These quantities are not used as primary evaluation metrics, but they provide a complementary view of the local behavior of the learned flow and the magnitude of policy-ratio fluctuations. As shown in Table C.9, ReFPO exhibits a lower Jacobian explosion rate, a lower two-sided log-proxy-ratio excursion rate, and smaller Jacobian norms than FPO.

Table C.9: Jacobian and policy-ratio diagnostic statistics on MuJoCo Playground.

Method	Jac. Expl.	Ratio Excursion	Mean Jac. Norm	Jac. Norm P95
ReFPO	0.005654	0.00189	0.4881	0.9067
FPO	0.008724	0.00365	0.5551	1.219

D Algorithmic details

Algorithm D.1 Reflow-regularized Flow Matching Policy Gradients (ReFPO)

Require: Policy parameters θ, value parameters ϕ, clip parameter δ, number of CFM draws M, proxy scale κ, Reflow coefficient λ, minibatch size B
while not converged **do**
  Collect rollouts $\{(o_i, a_i)\}$ with the current flow policy and compute advantages A_i
  $\theta_{\text{old}} \leftarrow \theta$
  for each rollout action a_i **do**
    Sample and store M noise/time draws $\{(\epsilon_{im}, \tau_{im})\}_{m=1}^{M}$
    Cache $\ell_i^{\text{old}} \leftarrow \ell_i(\theta_{\text{old}})$ by Equation 2, the κ-scaled average over the stored draws (no gradient)
  end for
  for each optimization epoch **do**
    for each minibatch $\mathcal{B}$ of size B **do**
      for each $i \in \mathcal{B}$ **do**
        $\ell_i \leftarrow \ell_i(\theta)$ by Equation 2 on the same stored draws {same definition as ℓ_i^{old}}
        $r_i \leftarrow \exp(\ell_i^{\text{old}} - \ell_i)$
        $L_i^{\text{FPO}} \leftarrow -\min\{A_i r_i,\ A_i \,\text{clip}(r_i, 1-\delta, 1+\delta)\}$
      end for
      $L_{\text{batch}} \leftarrow \frac{1}{B}\sum_{i\in\mathcal{B}} \left(L_i^{\text{FPO}} + \lambda \ell_i\right)$ {reuses ℓ_i; no extra forward pass}
      Form the PPO value loss $L_V(\phi)$ on the same minibatch
      Update (θ, ϕ) using $L_{\text{batch}} + c_V L_V(\phi)$, with PPO value coefficient c_V
    end for
  end for
end while

E Scope and derivations of the signed CFM analysis

Coupling and estimator. An on-policy action a_i is generated by the behavior-policy flow solver. The CFM proxy subsequently draws fresh ϵ_{im} and τ_{im} and stores them with the action. Its training pairs therefore have the policy action marginal but do not, in general, retain the original latent-to-action transport coupling. Classical Reflow instead pairs a source z_0 with the generated endpoint $\Phi_{\mathrm{old}}(z_0)$. The straight conditional target in Equation 2 is shared in form, while the coupling and signed optimization weights differ.

FPO reuses its stored draws and old loss across PPO epochs. Shared old/current draws provide a common-random-numbers comparison, while reuse across epochs also allows caching the old loss. Resampling fresh shared draws and reevaluating both policies is a valid alternative estimator but requires another frozen-policy evaluation. Fixing the draws is the implemented FPO estimator, not a mathematical prerequisite imposed by ReFPO.

Implementation conventions. The Playground implementation uses reversed time, $x_t = t\epsilon + (1-t)a$, with velocity target $\epsilon - a$. Denote its velocity predictor by w_θ. Setting $\tau = 1 - t$ and $v_\theta(x, \tau; o) = -w_\theta(x, 1-\tau; o)$ recovers the convention in Equation 2; both the velocity and its target change sign, preserving the squared residual. Equation 2 displays direct velocity supervision. FPO's noise-supervised variant reconstructs $\hat{\epsilon} = x_t + (1-t)w_\theta$, so $\hat{\epsilon} - \epsilon = (1-t)(w_\theta - (\epsilon - a))$. Thus its squared residual has a fixed per-draw weight $(1-t)^2$ relative to velocity supervision. The signed-gradient identity applies to this weighted proxy loss as well.

We separate raw, dimension-averaged CFM error C_i from the proxy scale, $\ell_i = \kappa C_i$. Playground uses $\kappa = 1$, whereas the Humanoid velocity proxy uses $\kappa = 1/(2 \cdot 0.05^2) = 200$. If the code adds $\lambda_{\mathrm{raw}} C_i$, the coefficient in Equation 4 is $\lambda = \lambda_{\mathrm{raw}}/\kappa$. Reported configuration coefficients refer to the raw implementation loss. This scaling preserves the signed-gradient argument but must be retained when comparing numerical thresholds or coefficients across domains.

Active and clipped branches. With $g_i = \nabla \ell_i$, differentiating $r_i = \exp(\ell_{i,\mathrm{old}} - \ell_i)$ gives $\nabla r_i = -r_i g_i$. Away from a clipping kink,

$$\nabla L_i^{\mathrm{FPO}} = c_i A_i r_i g_i, \tag{10}$$

$$\nabla\big(L_i^{\mathrm{FPO}} + \lambda \ell_i\big) = (c_i A_i r_i + \lambda) g_i. \tag{11}$$

On an active positive sample, FPO follows the same parameter-space integral curves as CFM gradient descent with a positive optimization-time reparameterization $du/ds = A_i r_i$, where u is distinct from the flow time τ. An active negative sample reverses that direction. For nonzero A_i and finite losses, $r_i > 0$, so the isolated FPO stationary points within an active branch coincide with those of ℓ_i. A CFM minimum with positive-definite Hessian lying inside that branch is attracting for positive advantage and repelling for negative advantage. On clipped plateaus the FPO gradient vanishes, while ReFPO still supplies λg_i.

For $A_i = -|A_i|$, the active ReFPO loss as a function of ℓ_i is

$$L_-(\ell_i) = |A_i| e^{\ell_{i,\mathrm{old}} - \ell_i} + \lambda \ell_i, \qquad \frac{\partial L_-}{\partial \ell_i} = -|A_i| r_i + \lambda.$$

Thus the regularizer attenuates anti-CFM growth, reverses it only when $\lambda > |A_i| r_i$, and can introduce a neutral point at equality. Increasing a variational CFM surrogate need not correspond one-to-one to decreasing exact clean-action log-likelihood; surrogate fit and variational tightness can both change.

Shared-field dynamics and occupancy. The per-sample coupling in Equation 5 and the batch restoring term in Equation 6 do not remove cross-sample interference. A realizable zero-residual solution for all pairs is a shared stationary point, but heterogeneous signed weights can create other cancellations. Equality of the complete stationary sets is not established for the full clipped, changing-data algorithm. Under an idealized KL-regularized policy improvement,

$$\pi_{k+1}(a \mid o) \propto \pi_k(a \mid o) \exp(\eta A_k(o, a)),$$

with $\eta > 0$, actions with larger advantage gain relative probability at a fixed observation. Changes in state–action occupancy additionally depend on the induced rollouts. This idealization motivates the occupancy-level interpretation in Section 3.2; it is not the exact FPO update and does not turn negative-advantage samples into Reflow samples.

Euler local truncation error. Let $x(\tau)$ solve $\dot{x} = v_\theta(x, \tau; o)$. Holding o and θ fixed, differentiating along the ODE yields $\ddot{x} = \partial_\tau v_\theta + J_x v_\theta \, v_\theta$. Taylor expansion gives

$$x(\tau + h) = x(\tau) + h \, v_\theta + \frac{h^2}{2}(\partial_\tau v_\theta + J_x v_\theta \, v_\theta) + O(h^3).$$

The Euler approximation omits the second-order term. The triangle inequality and $\|J_x v_\theta \, v_\theta\|_2 \leq \|J_x v_\theta\|_{\mathrm{op}} \|v_\theta\|_2$ give Equation 7. Here $O(h^3)$ denotes a remainder whose norm is bounded by $K|h|^3$ for small $|h|$. A sufficient condition is that v_θ is twice continuously differentiable near the solution segment and $x^{(3)}$ is bounded there. The same bound applies to backward steps. The CFM objective by itself does not control all derivatives in this bound, and the local expansion is not a global accuracy guarantee for a single step of length one.

Discrete-step ratio motion. For a shared-network minibatch gradient-descent step,

$$\Delta \log r_i \simeq \frac{\eta}{B} \sum_j (c_j A_j r_j + \lambda) \langle g_i, g_j \rangle,$$

the discrete-step analogue of Equation 5 with step size η. Only the self inner product is necessarily nonnegative. Cross terms can move an individual ratio in either direction, and the $+\lambda$ self term increases its immediate upward movement. The residual-based sensitivity derivation and the distinct upper- and lower-tail conditions are given in Equations 8 and 9. The discrete-step expression above is first-order in η and describes plain gradient descent; momentum and preconditioning alter the update direction. These local relations do not prove suppression of all ratio excursions.

F Details of the sign-conditioned intervention

Section 4.3 defines the intervention and reports its aggregate results in Table 3. The sign-masked terms $L_R^{\pm}$ both retain the full-batch denominator B, so adding L_R^- leaves every positive-sample weight unchanged; the sole intervention is the negative-sample regularizer on top of the common policy loss L_{PG}. The five paired BallInCup seeds are 0–4. For each seed, the positive-only and both-sign arms share the checkpoint, rollout batch, CFM draws, Adam state, and minibatch order, and each performs the normal 16 inner epochs at $\lambda = 0.04$. The disjoint held-out next rollout is collected by the unchanged pre-update policy and contains 983,040 transitions per seed, none of which is used for optimization. Table 3 reports mean $\pm$ SE over the five seed-level values; these SEs differ from the SDs in the main benchmark tables.

Each improvement direction in Table 3 holds in all five paired seeds. Under the sign-test null of independent, equally likely positive and negative differences, five same-sign nonzero effects give a two-sided p-value of $2(1/2)^5 = 0.0625$. This sign-only calculation does not determine the result of tests that also use effect magnitudes, and the study does not establish long-horizon performance.

For a concrete seed-4 illustration, a held-out positive-TD state at $\tau = 0.307$ has positive-only velocity residual $(-0.1821, 2.0065)$. Adding L_R^- changes its velocity by $(-0.000139, -0.001732)$, with correction cosine 0.985 toward the negative residual. The dimension-averaged squared error decreases from 2.029689 to 2.026240; the displayed vectors are rounded. Seven of eight sampled interpolation times on that chord are error-reducing. In a matched 480-step simulator rollout with common initial state and noise, the ball remains in the cup for 442 steps under positive-only regularization and 455 steps under both-sign regularization. This is an illustrative trace, not an aggregate return estimate.

G Targeted late-stage coefficient examples

The main MuJoCo results use a common $\lambda = 0.04$. To investigate visible late drops, additional BallInCup and CartpoleBalance runs used $\lambda = 0.03$ and 0.07, respectively, under the Figure 3 training protocol. Table G.10 preserves the actual logged evaluation checkpoints. The original and alternate-coefficient runs use different seeds, so these are illustrative trajectories, not a paired-seed coefficient ablation or a task-tuned benchmark result. Coefficient and seed effects are not separated.

The alternate-coefficient examples end at higher returns and show smaller peak-to-final drops. A controlled estimate of the coefficient's contribution would require matched seeds and a prespecified selection protocol; the examples do not replace the shared-coefficient main results.

Table G.10: Individual evaluation-return trajectories. BallInCup uses seeds 314 ($\lambda = .04$) and 404 ($\lambda = .03$); CartpoleBalance uses seeds 72 ($\lambda = .04$) and 222 ($\lambda = .07$). No error bars or cross-seed means are implied.

Steps (M)	BallInCup .04	BallInCup .03	Cartpole .04	Cartpole .07
0	0.0	0.0	330.6	317.5
11	13.5	0.0	440.9	441.9
22	655.2	253.0	807.3	821.2
33	810.5	810.2	961.2	950.2
44	886.0	907.0	978.5	965.1
55	896.9	909.8	986.1	972.3
66	939.3	863.8	984.1	957.9
77	624.1	808.7	570.6	972.0
88	665.2	875.2	284.5	991.4
100	518.3	815.5	369.0	985.2

H Positioning among recent flow-policy methods

Table H.11: Mechanism and setting comparison. NFE denotes velocity-field evaluations during action generation. Published returns are not compared across incompatible environments, data sources, and candidate-selection schemes.

Method	Optimization setting	Additional machinery	One-step relation
FPO++ [25]	On-policy flow PG	Per-sample ratios; asymmetric trust region	Stronger FPO backbone
NCDPO [24]	Noise-conditioned policy optimization	Backpropagation through denoising; behavior cloning	Distinct likelihood-based update
TRFP [29]	Off-policy MaxEnt RL	Replay, Q critics, hybrid ODE/SDE, truncation	Explicit flow straightening
FLAME [9]	Off-policy MaxEnt RL	Q reweighting, replay, entropy estimation	MeanFlow variant at NFE=1
FMER [5]	Off-policy flow RL	Weighted CFM, Q critics, entropy constraint	Does not establish forced NFE=1 results in the compared version
FPMD [3]	Off-policy mirror descent	Replay and Q critic	Flow/MeanFlow one-step policies
OFP [8]	Demonstration-based imitation	Self-distillation, consistency/guidance, warm start	One-step visuomotor policy
ReFPO	On-policy FPO	Reuses the existing CFM proxy residual	Evaluated at NFE=1 and 10

ReFPO retains FPO's on-policy value function without adding a Q critic, replay buffer, action-candidate selection, teacher, or separate distillation stage. Off-policy alternatives use critic-guided improvement and sample reuse, sometimes with entropy-related density estimation. Unlike them, ReFPO targets the PPO old/new ratio. Reusing flow evaluations does not guarantee lower total training cost.

I Implementation of the additional loss

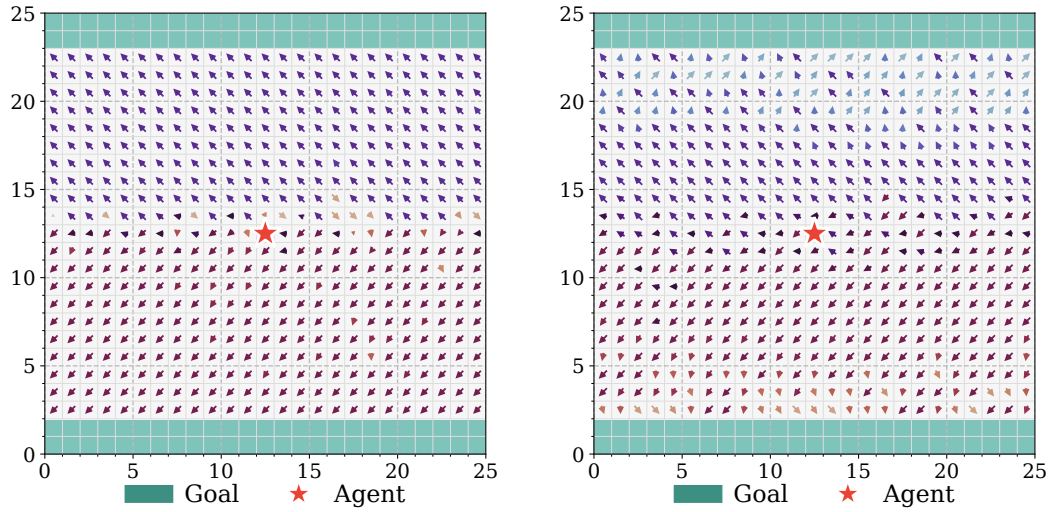
The additional loss reuses the unweighted current-policy CFM residual:

```
total_loss = (policy_loss + v_loss
              + self.config.reflow_loss_coeff * reflow_loss_val)
```

Here `reflow_loss_val` averages the raw, dimension-averaged current-policy CFM errors C_i; `reflow_loss_coeff` is λ_{raw}. Thus the term is $\lambda B^{-1} \sum_i \ell_i$ with $\lambda = \lambda_{\mathrm{raw}}/\kappa$ (Appendix E). It reuses the FPO residual, requiring no extra flow-model forward pass or differentiation through rollout sampling, but still incurs arithmetic, memory, and optimizer costs.

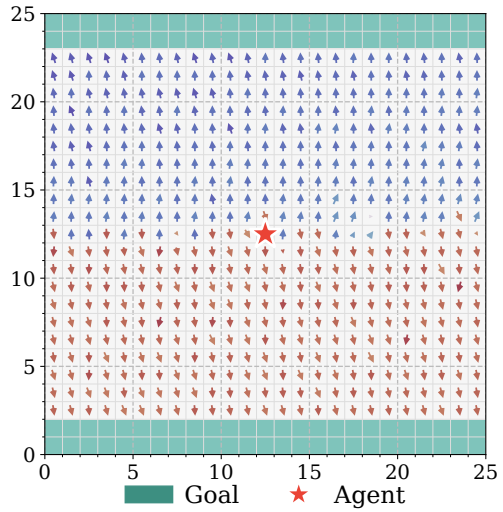
J Enlarged GridWorld action fields

The following panels enlarge the original sampled action fields without changing their underlying values. They show final sampled actions as a function of environment state, not internal ODE velocity along a noise-to-action path. The environment-state rollouts in Figure 2 should therefore be interpreted separately from the CFM geometry diagnostic.

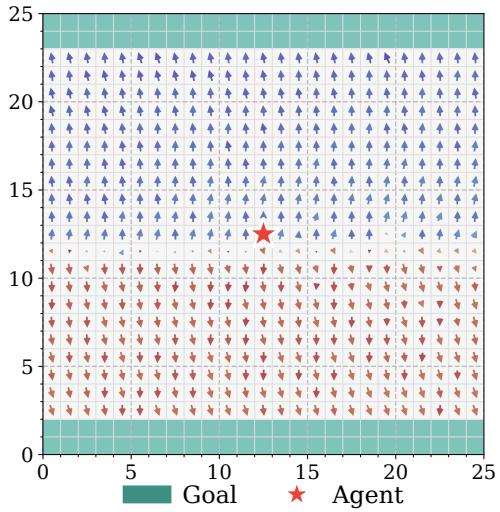


(a) FPO, ten steps.

(b) FPO, one step.



(c) ReFPO, ten steps.



(d) ReFPO, one step.

Figure 6: Enlarged versions of the submitted GridWorld sampled action fields. Direction and scale are preserved from the original vector graphics. Internal flow straightness is not defined by the orientation of these action fields or by vertical environment-state rollouts.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the main contributions of the paper, including the Reflow interpretation of FPO, the proposed ReFPO algorithm, and the empirical evaluation. They also describe the intended scope of the method for online flow-matching policy optimization.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses limitations in the final Discussion and Limitations section, including the local nature of the analysis, the focus on FPO-style online flow policies, and the dependence on the regularization coefficient.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The paper provides explicit derivations for the theoretical claims used in the method: the signed local CFM interpretation on active/clipped branches and the fixed-batch sensitivity analysis of the CFM proxy ratio. The assumptions and scope of these analyses, including the stop-gradient old policy loss and fixed-batch inner-loop setting, are stated in the corresponding sections and further discussed in the limitations.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The supplementary materials provide ReFPO training, configuration, and paired $N = 10/N = 1$ evaluation code. The method sections and Appendices B and D describe the algorithm and benchmark settings.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The supplementary materials provide the ReFPO training, configuration, and paired $N = 10/N = 1$ evaluation code and usage instructions. Benchmark data and licensed humanoid assets are obtained from the original providers, as described in Appendix A.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental settings are described in the main text, and the appendix provides detailed hyperparameters and training configurations for GridWorld, MuJoCo Playground, and Humanoid Control.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: The benchmark tables report cross-seed SDs, with taskwise aggregation specified for MuJoCo and full-motion-set seed metrics for Humanoid. The five-paired-seed intervention explicitly reports SE, while the targeted late-drop examples are labeled as individual runs with different seeds (Section 4.3, Table 3, and Appendices F and G).

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: Appendix B reports hardware and approximate per-run costs for the original benchmarks. Complete project-wide compute accounting, including the additional comparisons, interventions, and exploratory runs, is unavailable.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics. It uses standard simulated control benchmarks and does not involve human subjects, sensitive personal data, or deceptive data collection.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This work is a theoretical and algorithmic study of reinforcement learning for simulated control, and it does not introduce a direct deployment pathway or application-specific societal impact.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release high-risk models or datasets, such as pretrained language models, image generators, or scraped datasets with substantial misuse risk.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper credits the original simulators, codebases, and datasets. Appendix A identifies the upstream licenses and access restrictions, including the non-commercial research terms for AMASS and SMPL; those assets are obtained separately from their providers.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The contributed asset is the ReFPO implementation, with training and evaluation instructions alongside the code; the algorithm and loss addition are documented in Appendices D and I. No new dataset or pretrained model is released.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The work does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The work does not involve human-subject research, so IRB approval or equivalent review is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not used as an important, original, or non-standard component of the core methodology, scientific analysis, or experimental results.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.